AD-A007 586

SPEECH UNDERSTANDING SYSTEMS

William A. Woods, et al

Bolt Beranek and Newman, Incorporated

Prepared for:

Office of Naval Research
Advanced Research Projects Agency

March 1975

Unclassified
Security Classification

AD-A00[illegible]76

DOCUMENT CONTROL DATA - R & D

(Security cl. sif. tion of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING A (Corporate author)
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, MA 02138

2a. REPORT SECURITY CLASSIFICATION
none

2b. GROUP

3. REPORT TITLE
SPEECH UNDERSTANDING SYSTEMS
Quarterly Technical Progress Report No. 1
1 November 1974 to 1 February 1975

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Technical (1 November 1974 to 1 February 1975)

5. AUTHOR(S) (First name, middle initial, last name)
William A. Woods, Richard M. Schwartz, John W. Klovstad, Craig C. Cook, Jared J. Wolf, Madeleine A. Bates, Bonnie L. Nash-Webber, Bertram C. Bruce, and Victor W. Zue.

6. REPORT DATE
March 1975

7a. TOTAL NO. OF PAGES
~~70~~ 77

7b. NO. OF REFS
10

8a. CONTRACT OR GRANT NO.
N00014-75-C-0533

b. PROJECT NO.

c. Order No. 2904

d. Program Code No. 5D30

9a. ORIGINATOR'S REPORT NUMBER(S)
BBN Report No. 3018

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
A.I. Report No. 24

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
ONR
Department of the Navy
Arlington, VA 22217

13. ABSTRACT

This report covers research and development work done from 1 November 1974 to 1 February 1975 under the Speech Understanding Systems Contract No. N00014-75-C-0533. Areas included in this work are acoustic-phonetics, lexical retrieval, lexical verification, and natural language syntax, semantics and pragmatics. The report consists of two parts -- a brief Survey of Progress containing a few paragraphs describing the major progress in the individual components of the project, and a Technical Notes section containing detailed specifications of experiments performed, programs implemented, design studies, and, where appropriate, supporting data and appendices. This first QPR contains such technical notes on acoustic segmentation and labeling, lexical retrieval, the syntactic component and the semantic network utility package.

Unclassified
Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Acoustic-phonetics						
Acoustics						
Acoustic Segmentation						
Database						
Dates						
Dip Detector						
Discourse Structure						
Duration of Fricatives						
Encoding Contextual Dependence						
Grammars						
Hand Labeling						
Intention						
Labeling						
Lexical Retrieval						
Parameter Reading						
Parsing						
Phonetic Baseforms						
Phonetic Representations						
Phonological Rules						
Phonology						
Place of Articulation of Unvoiced Plosives						
Place of Articulation of Strident Fricatives						
Pragmatics						
Probabilistic Segment Specification						
Scoring Alternatives						
Scoring Philosophy						
Semantic Networks						
Semantics						
Tree Search						
Tree Structures						
Verb-Particle						
Voice Onset Time (VOT)						

Security Classification

PREFACE

This Quarterly Progress Report (QPR) marks the first QPR of a new speech contract separate from the larger ARPA contract (DAHC15-71-C-0088) under which speech research has formerly been performed.

Since we will now be issuing QPR's under a new contract and for a new contract monitor, I would like to adopt a slightly different editorial policy from that which we have followed in the past. Each QPR in the new series will consist of two parts -- a brief Survey of Progress containing a few paragraphs describing the major progress in the individual components of the project, and a Technical Notes section containing detailed specifications of experiments performed, programs implemented, design studies, and, where appropriate, supporting data and appendices. The Technical Notes will aspire to publication quality although they will in general assume knowledge of the continuity of the project and thus lack much of the introductory material which would be present in a self contained publication. They may also include appendices and tables of supporting data in excess of that which would be permitted in most journal publications. It is hoped that the Technical Papers section will serve as an archive of information which has been discovered in the course of the project that will be of use for other researchers.

The Managing Editor for the new QPR series is Ms. Bonnie Nash-Webber.

W.A. Woods

Table of Contents

I. PROGRESS OVERVIEWS

A. Acoustic-Phonetics

During the past months, we have been putting a great deal of effort into the design and preliminary implementation of parameter-based segmentation and labeling strategies. These have been based on intuitions developed through a continuing series of organized parameter reading sessions, which have had the additional benefit of providing us with segment lattices for use in lexical retrieval experiments. Section II of this report will present a summary of these sessions and their results to date. It will also describe the preliminary segmentation programs which have been based on these results and the improvements to our labeling algorithms also deriving from them.

B. Lexical Retrieval

Recent work on the lexical retrieval component has consisted of the formulation, implementation, and extension of our scoring philosophy and lexical lookup procedure, along with corresponding work on our lexicon. The scoring philosophy is based on Bayesian analysis and involves finding the most probable utterance and pronunciation model for a given acoustic waveform. The new lexical lookup procedure, designed to handle alternate pronunciations, segmentation errors and boundary effects in a fast and efficient way, has resulted in a restructuring of the lexicon into a tree format. Section III of this report presents the arguments for and details of the work done in these areas.

C. Verification

Work in word verification during the past quarter has been directed towards improving the quality of the synthesis. In order to give the Verification component access to phonological knowledge, we have broadened the phonetic context in which the phonetic-to-acoustic parameter conversion is done to include a generative phonological mechanism. This was done by embedding the conversion program in the Bobrow-Fraser rule tester formalism [1] and adding features with associated numerical values to the existing set of binary distinctive features. In addition, we have worked on expanding and modifying the set of phonological rules to deal with these new features.

In order to judge the quality of synthesis programs, we have also implemented a waveform synthesizer which accepts a parametric representation as input. This component is currently being tested using parameters mechanically extracted from actual utterances. These will serve as benchmarks against which we can compare the parametric output of successive implementations of the rule-driven phonetic-to-acoustic synthesizer.

D. Hardware

During the past quarter, we took delivery of two small computer systems which will form a signal processing facility for both the speech understanding and speech compression projects, a DEC PDP11/40 and a Signal Processing Systems Inc. SPS-41 (which was purchased on a previous contract).

The PDP11 has 32K of parity core memory and memory management and extended arithmetic options. This is augmented by 24K of Standard Memories core, 8K of semiconductor memory shared with the SPS-41, and a Telefile DC16H/CD213 disk. The disk system has a capacity of 30 million words; although it is moving-head, it is sufficiently fast to support spooling to and from the A/D and D/A interface at rates in excess of 20,000 samples per second. Our IMLAC graphics system will also be connectible directly to the PDP11.

The SPS-41 signal processor is connected to the PDP11 UNIBUS; its most efficient data communication path with the PDP11 is via the 8K semiconductor shared memory mentioned above. The SPS-41 contains an Input-Output Processor of exceptional versatility, so our machine has dual 12-bit A/D and D/A converters, together with the necessary clock hardware installed there. In addition to playing out sampled signals, the D/A's are also valuable debugging aids, for they may be used to drive oscilloscope displays of data buffers in the SPS-41 at various stages of signal processing.

We do not yet have an ARPANET interface for the PDP11, and we are still awaiting implementation and documentation of the virtual memory ELF operating system for the PDP11. The lack of a file system in ELF is also an obstacle; we will probably have to implement a temporary one in user code for use until one is developed for the ELF system.

E. Syntax

The grammar for the syntactic component has been both expanded to include a sub-grammar for parsing date expressions and hand verb-particle constructions, and also simplified to build more easily interpretable structures for previously accepted utterances.

A system of weights on the arcs to the grammar has been developed to allow the parser to score parse paths relative to one another in order to choose the best set of paths for extension. Experimentation is underway to explore the various mixtures of depth first and breadth first parsing strategies which are now available to the parser.

Section IV of this report gives details and examples of this work.

F. Semantics

As well as continuing in the construction of a semantic network to represent the conceptual structure underlying the travel budget management lexicon and in the parallel development of functions for semantic theory building which understand additional types of semantic network relationships, work on semantics has been directed to the extension and improvement of the basic network formalism. The result has been the production of a set of general semantic network utility packages for creating, editting, accessing, printing and merging semantic networks. This will be described in more detail in Section V.

G. Pragmatics

The Pragmatics component is currently being developed to perform four functions: to complete the interpretation of a theory on the basis of pragmatic information, to *evaluate* a thus completed interpretation, to make suggestions to semantics and syntax and to *execute* a complete utterance interpretation. In all cases, procedures are involved to apply knowledge about the discourse and the intention of the speaker.[2]

The input to Pragmatics is a theory word list, a partially instantiated case frame token from Semantics plus the corresponding structure from syntax. The first procedure (INSTANCE-MAP) determines likely intentions on the basis of words in the theory, e.g. a simple declarative statement probably implies "add new information" or "edit old information". The next procedure (MODE-STATUS) suggests possible intentions on the basis of the discourse structure. The resulting intentions are then combined and used by the procedure REIFY to fill in ellipsis and resolve anaphoric references. Completing an interpretation also involves performing quantifier scoping. This is done by LIFT-QUANT which moves inner quantifiers to the outermost position. Finally, either EXECUTE or EVALUATE is called on the completed interpretation. EXECUTing an interpretation may add, delete, or change the data base; or retrieve information. Accordingly, data base operations are directly under the control of Pragmatics. EVALUATing an interpretation results in a list of case-score-suggestion triples for each case in the case frame taken, as well as a score and suggestion list for the token as a whole.

Scores are discrete valued indicators of the likelihood of either a particular case filler or case frame token. Suggestions are either substitutes for unlikely case fillers, proposals for likely ones, or higher concepts in which the concept expressed by the given case frame token may be embedded. E.g. if a trip description refers to an existing trip, then "edit" is a likely higher concept.

These procedures are currently under development and will be described in detail in succeeding QPRs.

References

[1] Bobrow, Daniel G. and J. Bruce Fraser, "A Phonological Rule Tester",CACM Vol. 11, No. 11, pp. 766-772 (November 1968).

[2] Woods, W.A., M. Bates, B. Bruce, J. Colarusso, C. Cook, L. Gould, D. Grabel, J. Makhoul, B. Nash-Webber, R. Schwartz and J. Wolf, "Natural Communications with Computers Final Report - Volume I, Speech Understanding Research at BBN October 1970 to December 1974", BBN Report No. 2976 [Vol. I], Bolt Beranek and Newman Inc, Cambridge, Ma. (December 1974).

II. ACOUSTIC SEGMENTATION AND LABELING

A. Data Collection

During the past quarter we have expanded our data base of sentences related to the travel budget task. We now have 47 digitized utterances on line (from 3 male speakers) taken from a list of 27 sentences (See Appendix A). Twenty of these utterances have been carefully hand labeled, guided by parameters, spectrograms, time-waveforms, and original analog recordings. An ideal hand labeling indicates

1) The time (in 100 microsecond units) of the beginning of each phonetic element, each word (marked by "/"), and each syllable (marked as "*").

2) The silent period (SI) and the burst and aspiration of plosives (See Appendix B).

3) The stress levels assigned to vowels (0=unstressed, 1=secondary stress, 2=primary stress).

These utterances are being used to test our acoustic segmentation and labeling strategies. In addition our statistics gathering program is using them to generate quantitative statistical measures for later use by the segmentation and labeling programs. The existence of such a data base is crucial to the successful development of advanced segmentation and labeling algorithms, and we plan to continue its expansion in parallel with the development of our acoustic analysis module.

B. Parameter Reading

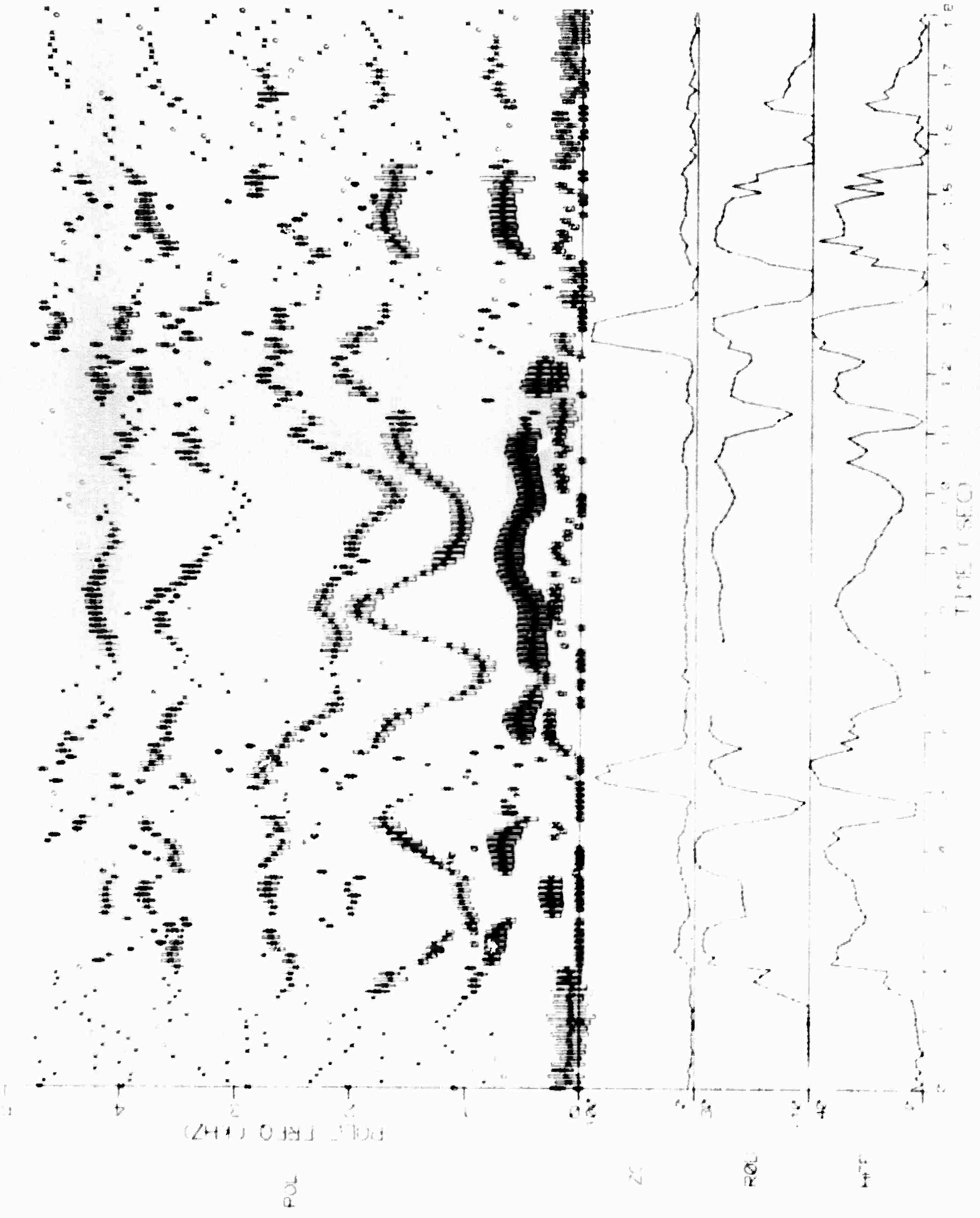
In order to gain insight into segmentation strategies, we have been holding organized parameter reading sessions over the last month, similar to earlier spectrogram reading sessions. With no *a priori* knowledge of the content of an utterance, the readers use a number of energy related parameters to segment the data into major categories: sonorants, vowels, strident or weak fricatives, and plosives. (See Figure 1 for a plot of these parameters for a sample sentence.) Obstruents are also classified as voiced or unvoiced if possible. Poles from the linear predication analysis of the utterance are then used to determine vowel and consonant identities based on steady state and transitional values, respectively. Five sample segment lattices resulting from the blind reading are shown plotted above the ideal transcription in Figures 2a-2e.

Our experience to date on parameter reading can be summarized as follows:

a) Human segmentation error is less than one percent, taking into account phonological variations which will exist in the lexicon.

b) The resulting segment lattice is never more than two deep, and alternate paths occur on the average, twice per utterance. (An average utterance consists of 25 segments.)

c) Because we were reading parameters, rather than spectrograms, we believe our strategies can be implemented with computer programs with relative ease. In fact, we believe that we can develop an acoustic segmenter to mimic human parameter reading well enough to yield comparable performance.


POL
POLE FREQ (KHZ)
0
1
2
3
4
5
20
ZC
0
30
RØL
-10
40
HFF
0
TIME (SEC)
0
.1
.2
.3
.4
.5
.6
.7
.8
.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8

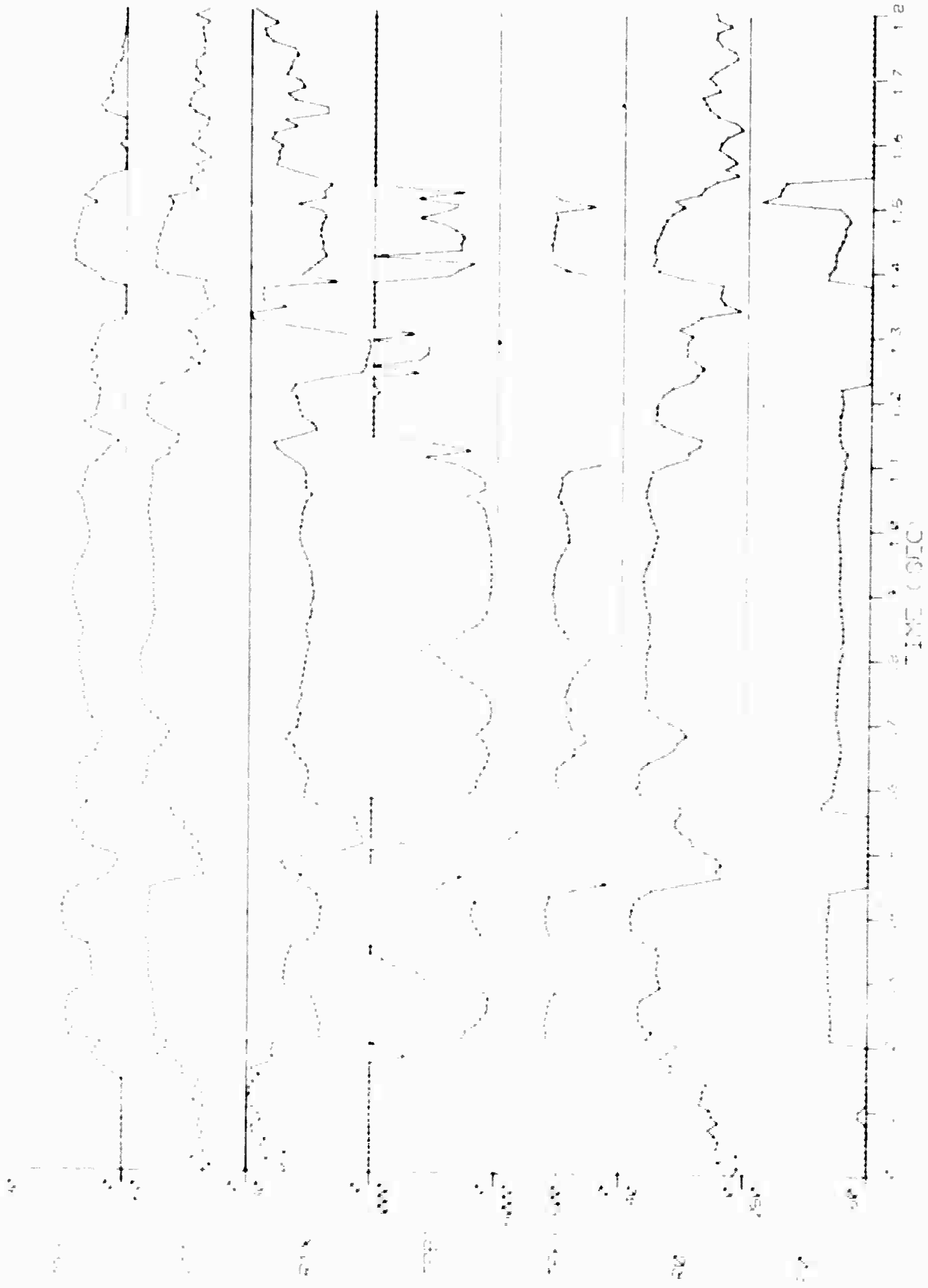


Figure 1: Parameters Used for Blind Reading

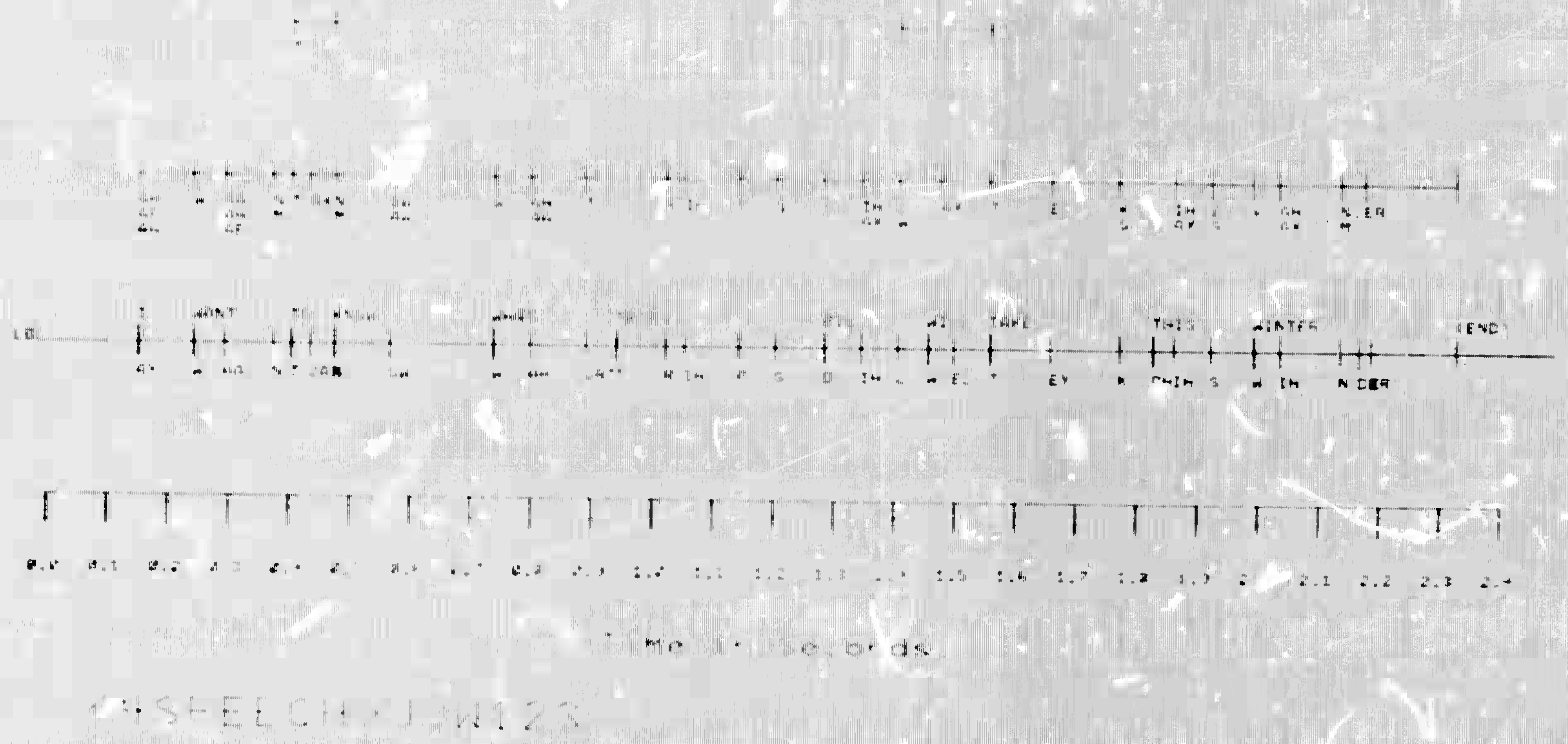
AAA126

d) Formant targets, transitions and context dependent acoustic phonetic knowledge were used extensively. A successful labeler must be able to incorporate such knowledge.

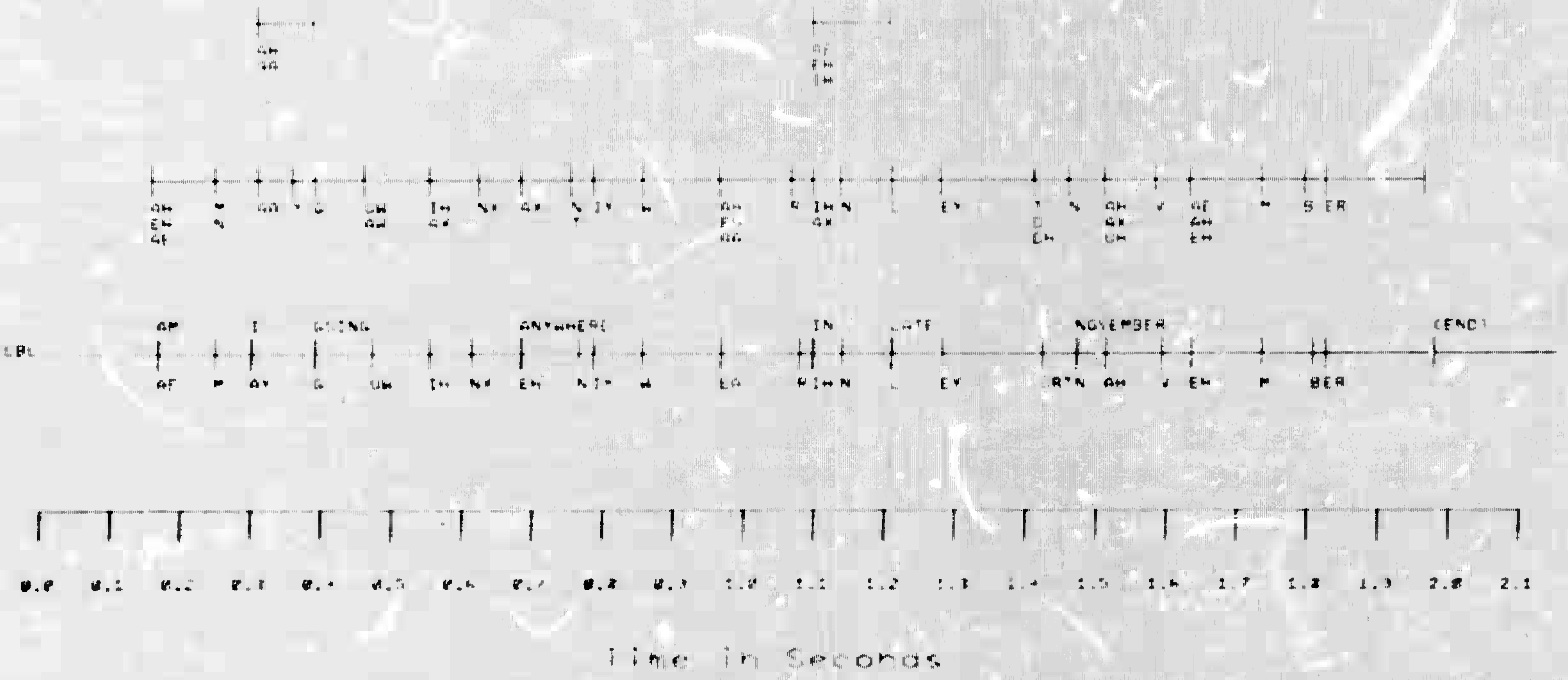
(A further benefit of these parameter reading sessions has been the resulting segment lattices, which are being used in lexical retrieval experiments. Also, performance analysis after the "blind" reading experiment results in a correct ideal transcription based on the lexical identity of the utterance. This is taken as the standard of correctness for all segmentation and labeling experiments and used in the data base for statistical measurements for the computation of lexical scores.

(END)
Time in Seconds
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
<4SPEECH>JM12

[Figure 2a]

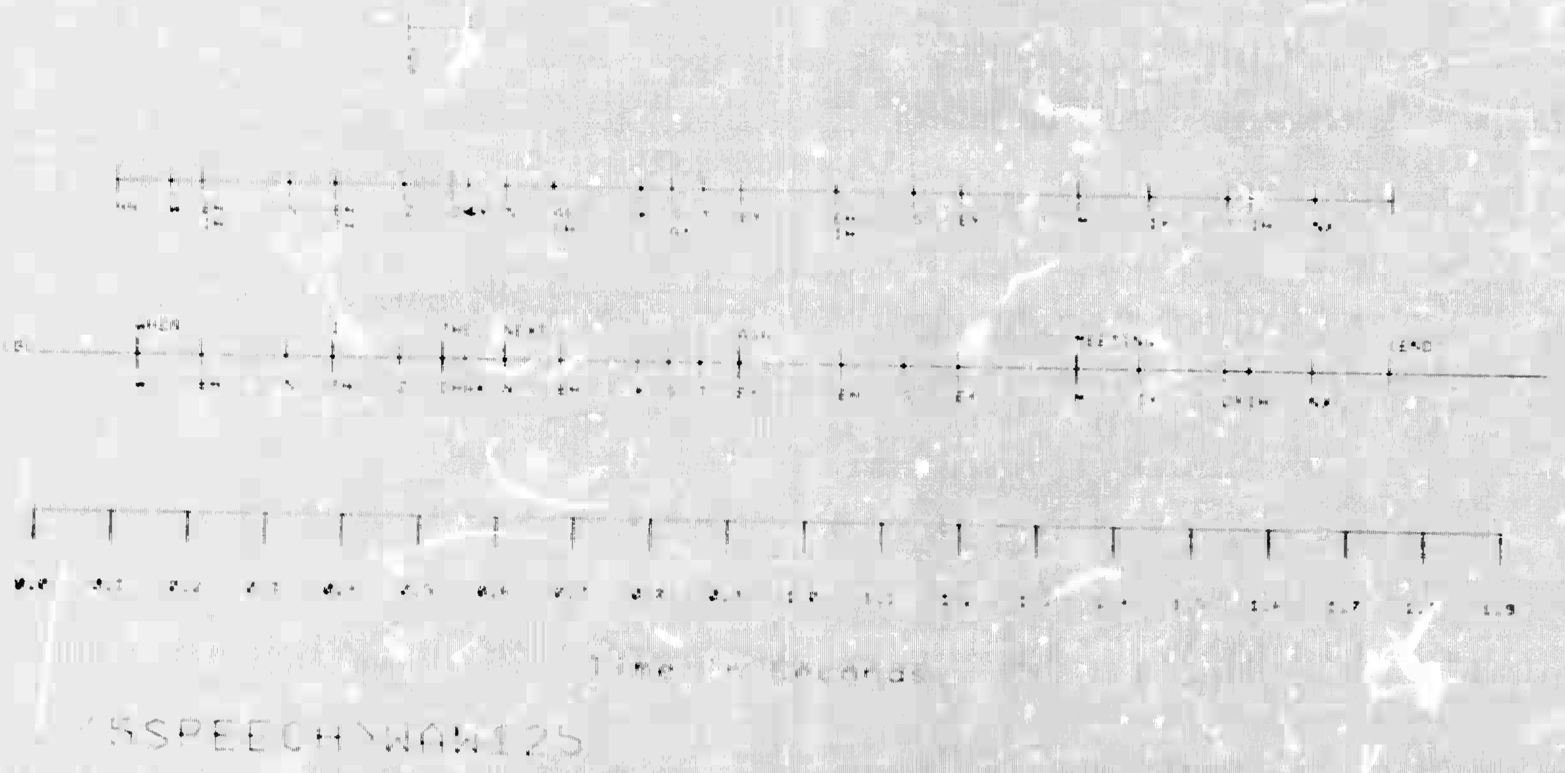

THIS
WINTER
(END)
EY
CHIH S
IH
N DER
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
Time in seconds
<SPEECH>J-W123

[Figure 2b]

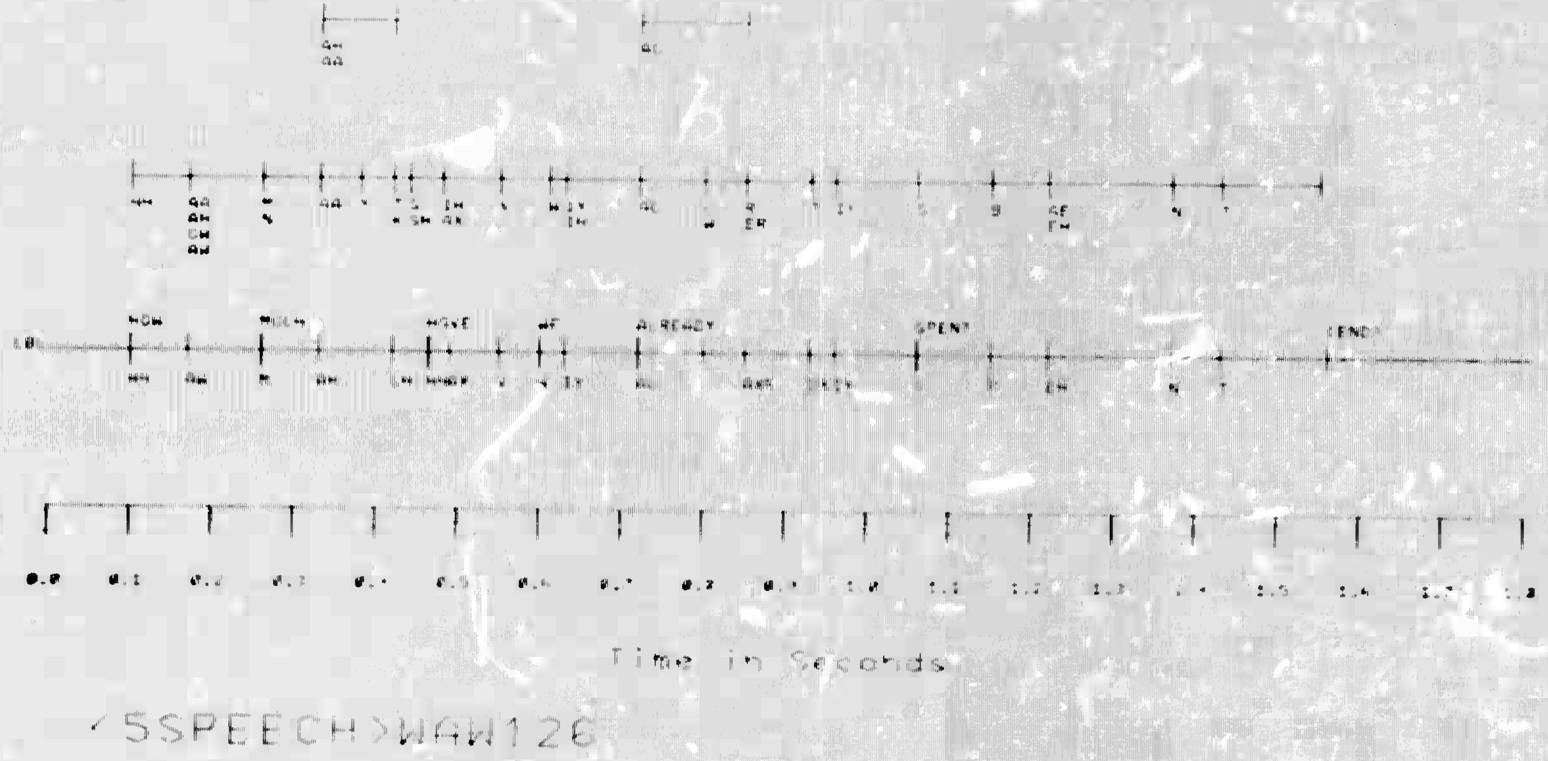


<(SPEECH>JJW124

[Figure 2c]


(B)
WHEN
I
THE
NEXT
ASK
MEETING
(END)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
Time in Seconds
SSPEECH>WAW125

[Figure 2d]


HOW
MUCH
HAVE
WE
ALREADY
SPENT
[END]
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
Time in Seconds
<SSPEECH>WAW126

[Figure 2e]

C. Preliminary Segmentation Programs

In an attempt to simulate the preliminary phase of the segmentation performed by human parameter readers, we developed a program which looks for boundaries between sonorant and obstruent sequences using the parameter LFE (low frequency energy from 120-440 Hz.). The global level of LFE fluctuates, usually decreasing, over an utterance, and obstruents frequently exhibit small but noticeable dips which have fairly high minima. Consequently, a general dip detector with several variable parameters was developed for looking at curves, and detecting dips and plateaus adjacent to dips. Its purpose is to locate all obstruents which have low energy in the low frequencies. This includes all unvoiced sounds, most occurrences of voiced plosives, all strident fricatives, and most occurrences of [V, DH, HH, DX]. (There are times when these latter obstruents occur between vowels that there is no dip in LFE; however, at these times a large decrease in energy in the higher frequencies can be noted.)

In the first test run of this program on 37 utterances (spoken by 3 male speakers, for a total of 1145 phonetic segments in 357 sonorant sequences and 383 obstruent sequences), there were 10 places where errors were made and incorrect dips were found. Six of these were in the last syllable of the utterance, where amplitude and fundamental frequency drop off; seven occurred in the 8 sentences spoken by a speaker with very low fundamental frequency (WAW). What follows is a more detailed discussion of the cause of these errors.

Since our analysis is not pitch-synchronous, the energy in the low frequencies can fluctuate rapidly when the pitch period becomes greater than 10 msec for a 20 msec analysis window. We have spent some effort trying to distinguish these fluctuations from those due to voiced plosives or weak fricatives. We are currently investigating the use of a zero-phase unit-gain filter to smooth out these effects. Unfortunately, this filter also eliminates some of the dips which should be found, but we hope to be able to find a reliable and computationally reasonable procedure for detecting this condition and eliminating this source of error.

In a second test run, the threshold for dips was increased slightly, to eliminate falsely detected dips, with the result that several of the correct dips were also missed. However, when this threshold was combined with the original one and dips found by only the lower threshold were treated as optional, only 3 errors remained. That is, 7 of the incorrectly found dips from the first test run were made optional. Hence in absence of a procedure to eliminate the above source of segmentation error, it appears we can deal with most of it by labeling questionable segmentations as optional.

Preliminary tests indicate that this program can also be used to find nasals and some glides within sonorant sequences (operating on energy from 640-2800 Hz.). It also may be useful in looking at other bands of energy.

D. Improvements to Statistics Package

A module was added to the display routines of the statistics package, enabling scatter diagrams to be made in 3 dimensions. With the aid of reference lines and the ability to rotate the display, it is possible to develop more complex decision spaces. It is now also possible to superimpose any combination of the previous 15 scatter diagrams or distributions (See Figures 3-5). The program has also been made faster and more flexible to improve interactions. Searching 20 utterances for a prescribed context and tabulating the desired statistics takes less than 1 second.

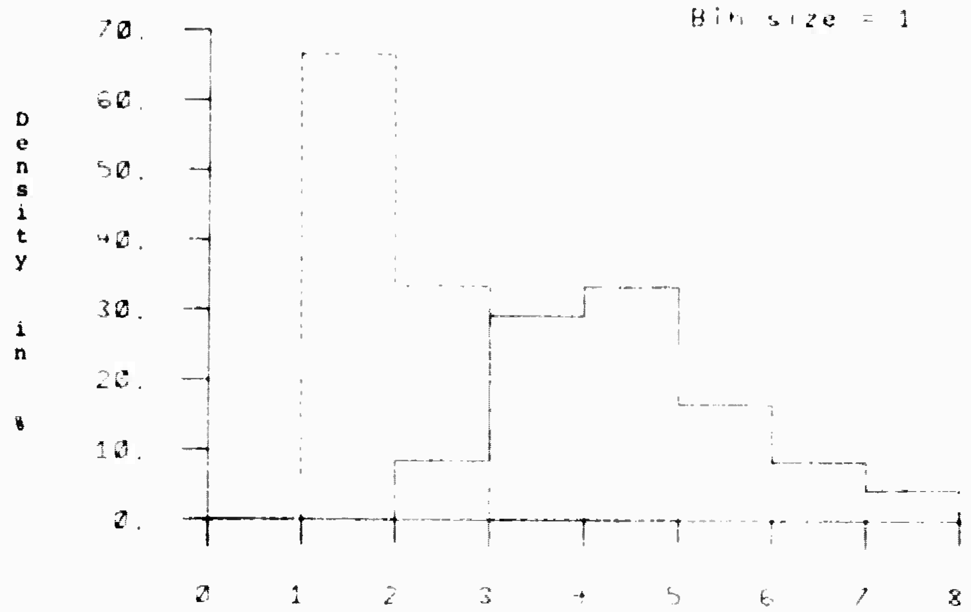
E. Acoustic-Phonetic Algorithms Developed

In reading parameters, we found that there were some segmentation and labeling decisions which were difficult to make. Therefore, we ran short experiments with the statistics facility to try to arrive at reasonable decision criteria for these cases. Several classification algorithms were also derived while developing new features of the statistics gathering facility. The following is a list of these difficult types of decisions and the criteria we set up for making them.

1) For plosives followed by vowels (but not preceded by strident fricatives), the voiced/unvoiced distinction was made by measuring a parameter related to voice onset time (VOT). Rather than using the VOT indicated in the ideal labeling, this period was determined by searching for the burst (indicated by the lowest 2nd derivative of energy after its minimum value) and the beginning of the vowel (indicated by the maximum derivative of energy after the burst). This more complicated procedure was used to ensure that measuring VOT automatically was possible.

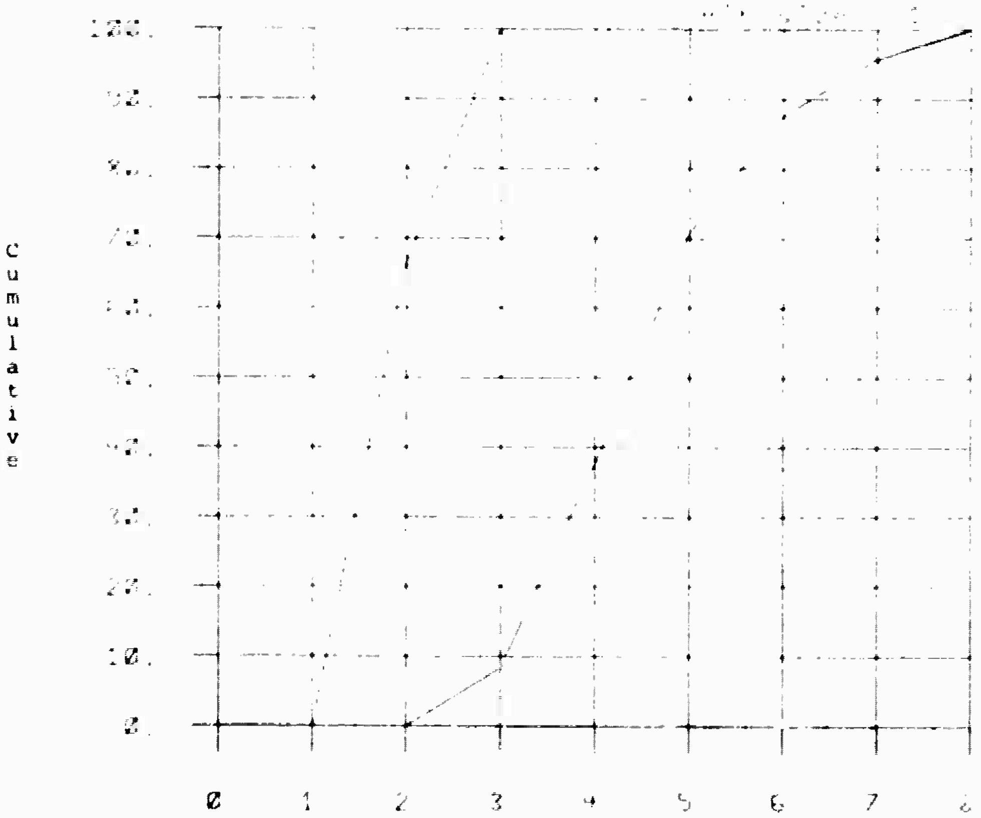
This duration correctly classified 46 of the 48 plosives examined as voiced or unvoiced. Figure 3a contains the density distributions for voiced (dotted line) and unvoiced (solid line) plosives. The time scale is in units of 10 msec frames. The region of overlap indicates that 8% of the 24 unvoiced plosives would be incorrectly classified as voiced, if a decision boundary were assigned just below 30 msec. (In fact, our basic philosophy precludes assigning decision boundaries whenever there is a nonzero overlap, but error rate is a good subjective measure of minimum performance.) The cumulative distributions shown in Figure 3b with grid lines superimposed illustrate another way of evaluating the performance of an algorithm.

Though this performance is good, it is felt that it can be improved. (Both errors were the result of an error in locating the burst.) The time measures used were rounded to the nearest 10 msec, but finer measures may improve performance. Also, dependencies on place of articulation and the following vowel and stress level were not considered.



VOT for Voiced vs Unvoiced Plosives

[Figure 3a]

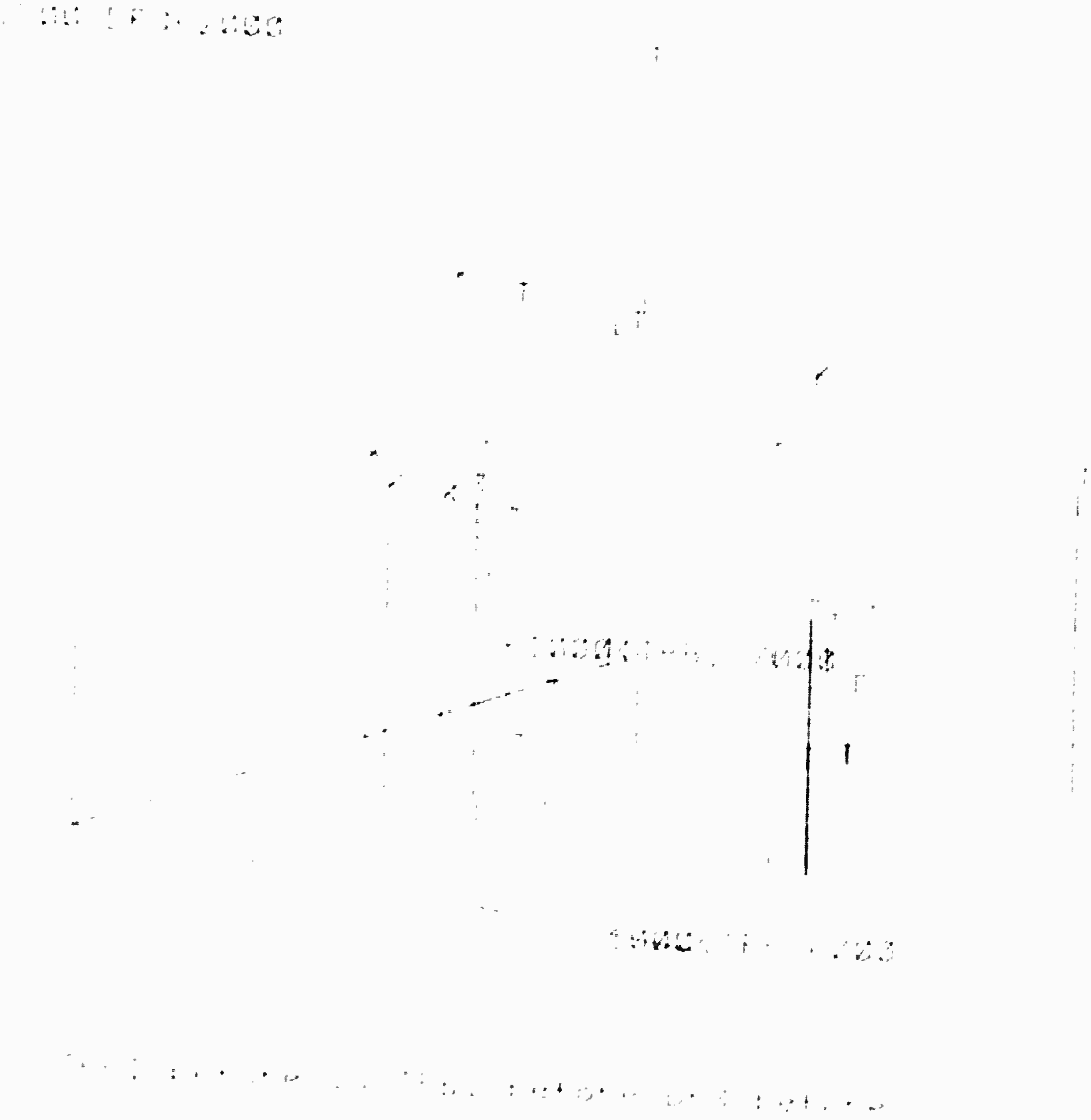


VOT for Voiced vs Unvoiced Plosives

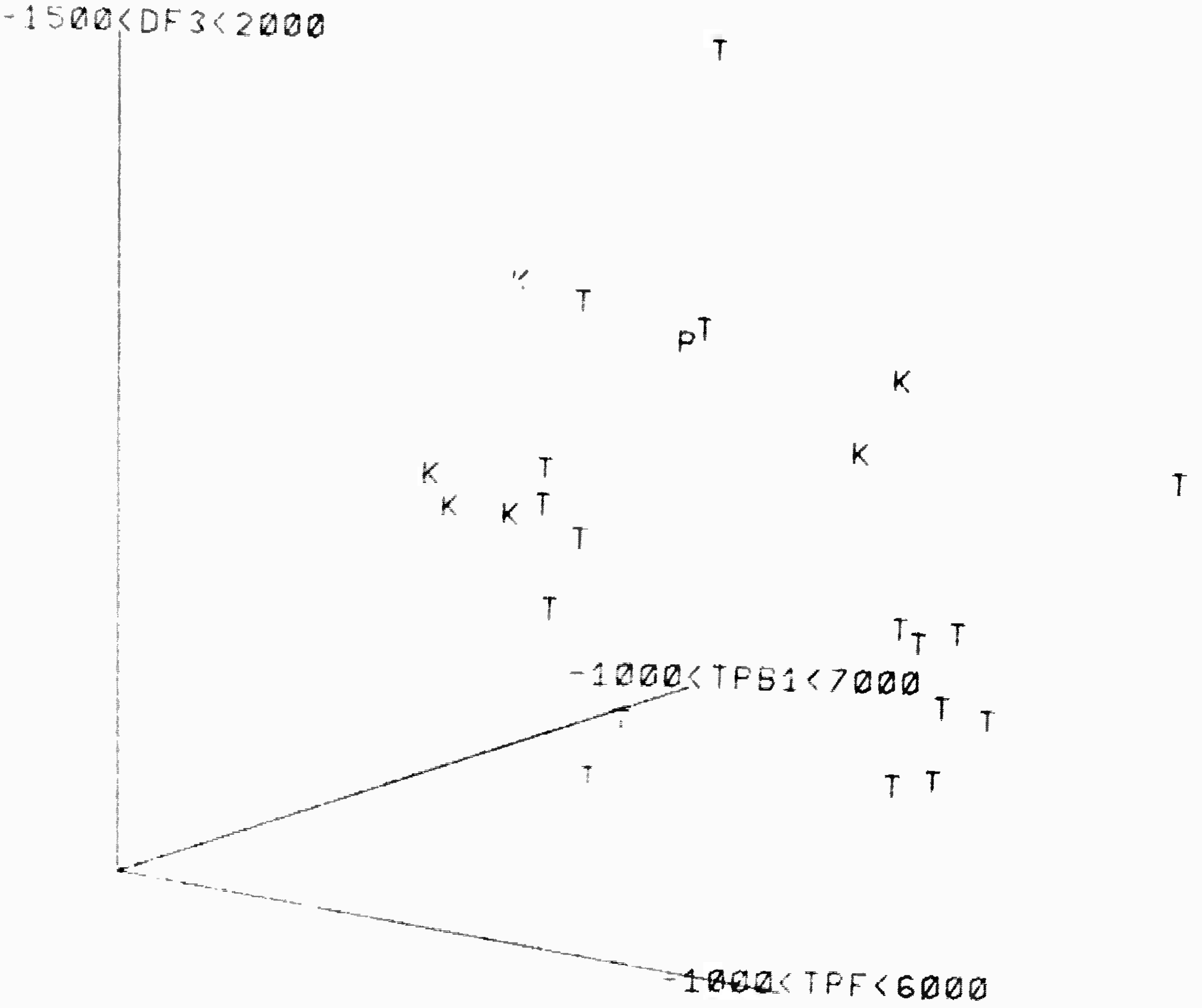
[Figure 3b]

2) For plosives followed by vowels, the place of articulation of the plosive was determined using the two-pole frequency approximation to the peak for the 20 msec analysis window centered around the burst. Also used is the 10 msec change in F3 just before the silence. (This could clearly be made more complex.) Figure 4a shows a two dimensional view of a scatter diagram in the 3 dimensions described, rotated to show maximum separation of classes. There are 6 [k]´s, 17 [t]´s and 1 [p] for speaker JJW. Reference lines are drawn from each data point - at the lower left of each label - to the plane DF3=0, to aid in visualization of their relative locations (Figure 4b shows the same plot without the reference lines.) When the two-pole procedure models the spectrum as 2 real poles, the frequencies of the poles are always 0 and/or 5000 Hz. For those [t] and [k] bursts which have a pole at 5000 Hz, (The lower ends of their reference lines form a straight line.) the other 2 parameters must be used.

The boundary separating the [t]´s and [k]´s in the group on the left might be questioned, since there are several samples of [t]´s and [k]´s which are quite close to each other. Though one would expect the frequency and bandwidth of a burst to be related, more data should be used to verify this boundary. Figure 4c is a view of the same data from the "top" of Figure 4a. This accentuates the group with a two-pole frequency at 5000 Hz. It also shows that most of the other [t]´s and [k]´s are separable by frequency alone. The [p] and [t] appear inseparable, so this would cause one error in a decision oriented system. Since more data is needed, and burst characteristics can be speaker dependent, 14 more samples were taken from speaker DWD. These are displayed with the initial samples in Figures 4d and 4e. Comparison reveals that all the samples (8) of [t] for DWD have a frequency of 5000 Hz and a consistently lower bandwidth than those for speaker JJW. Note (Figure 4e) that the frequency during the burst of a [p] in un-preemphasized speech is low due to the absence of any high frequencies. (The burst frequency is around 10-12 kHz, much past the 5 kHz range.) The group of 19 [t]´s and [k]´s at 5000 Hz appear harder to be separated, but it can be seen that, within the plane of TPF1=5000, the [t]´s form a semi-circle around the [k]´s. More data is needed. There is still only 1 definite confusion in the 38 samples - the [t] with TPF1=0.

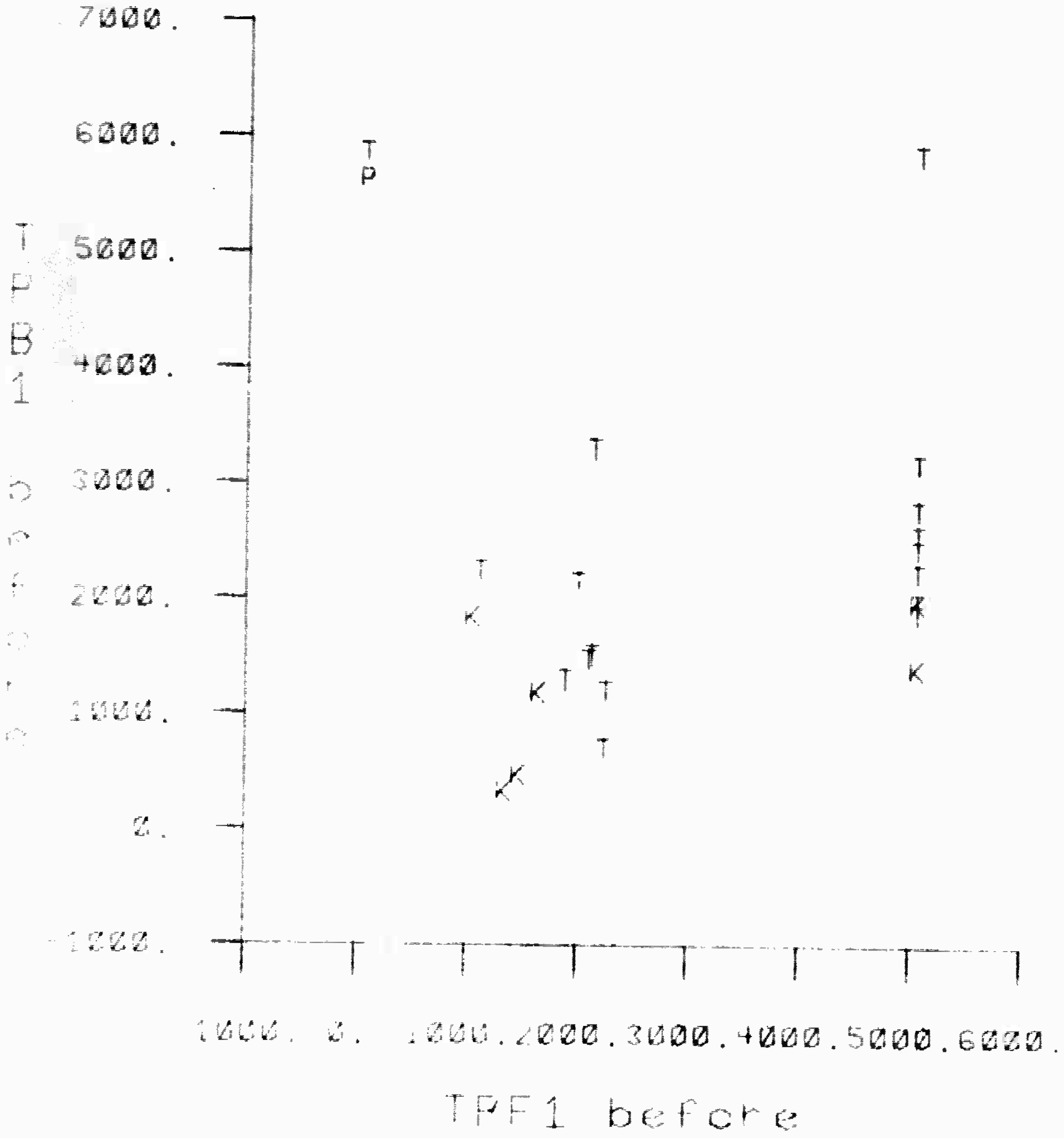


[Figure 4a]

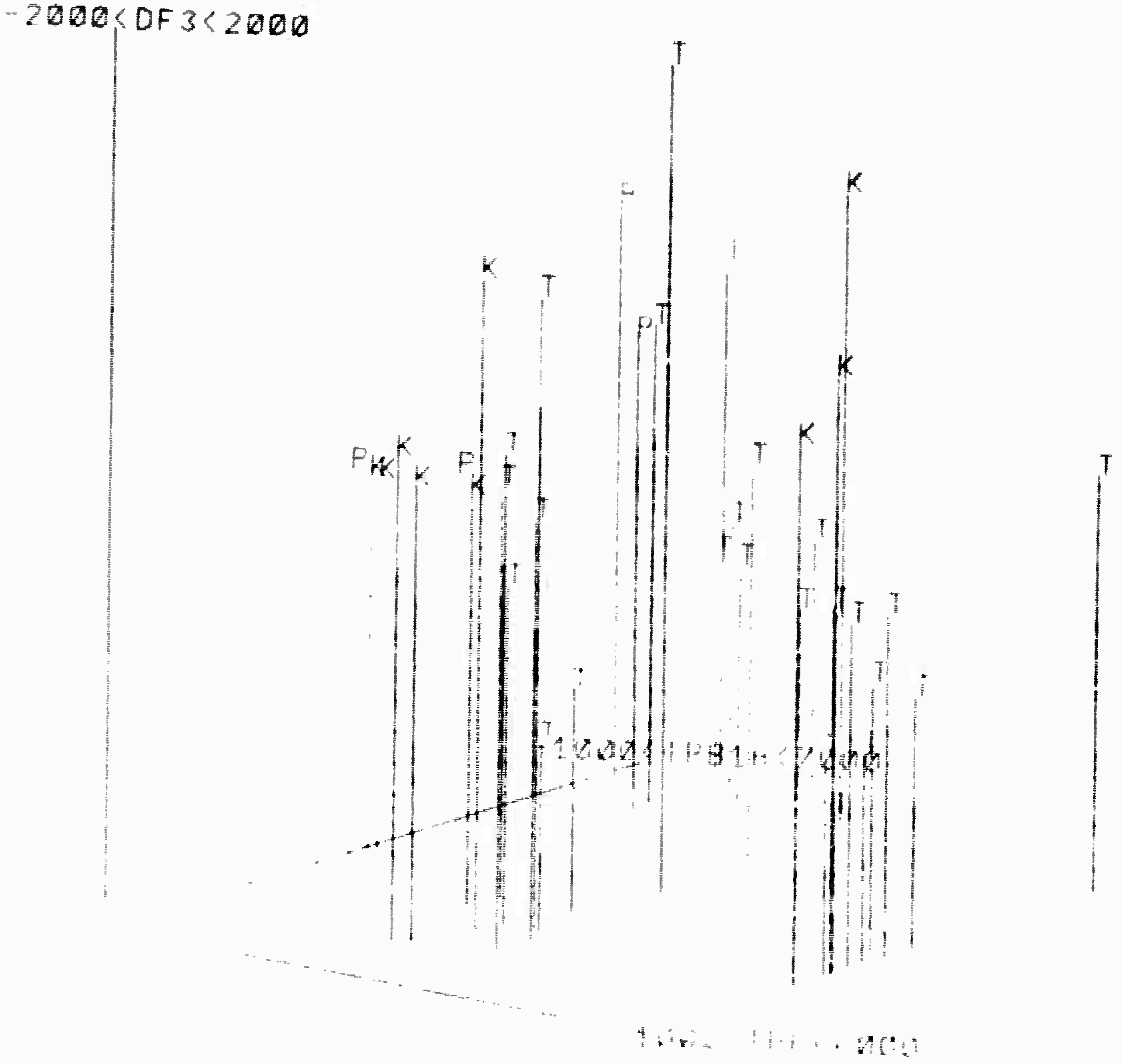


TPF1 before vs TPB1 before DF3 before

[Figure 4b]

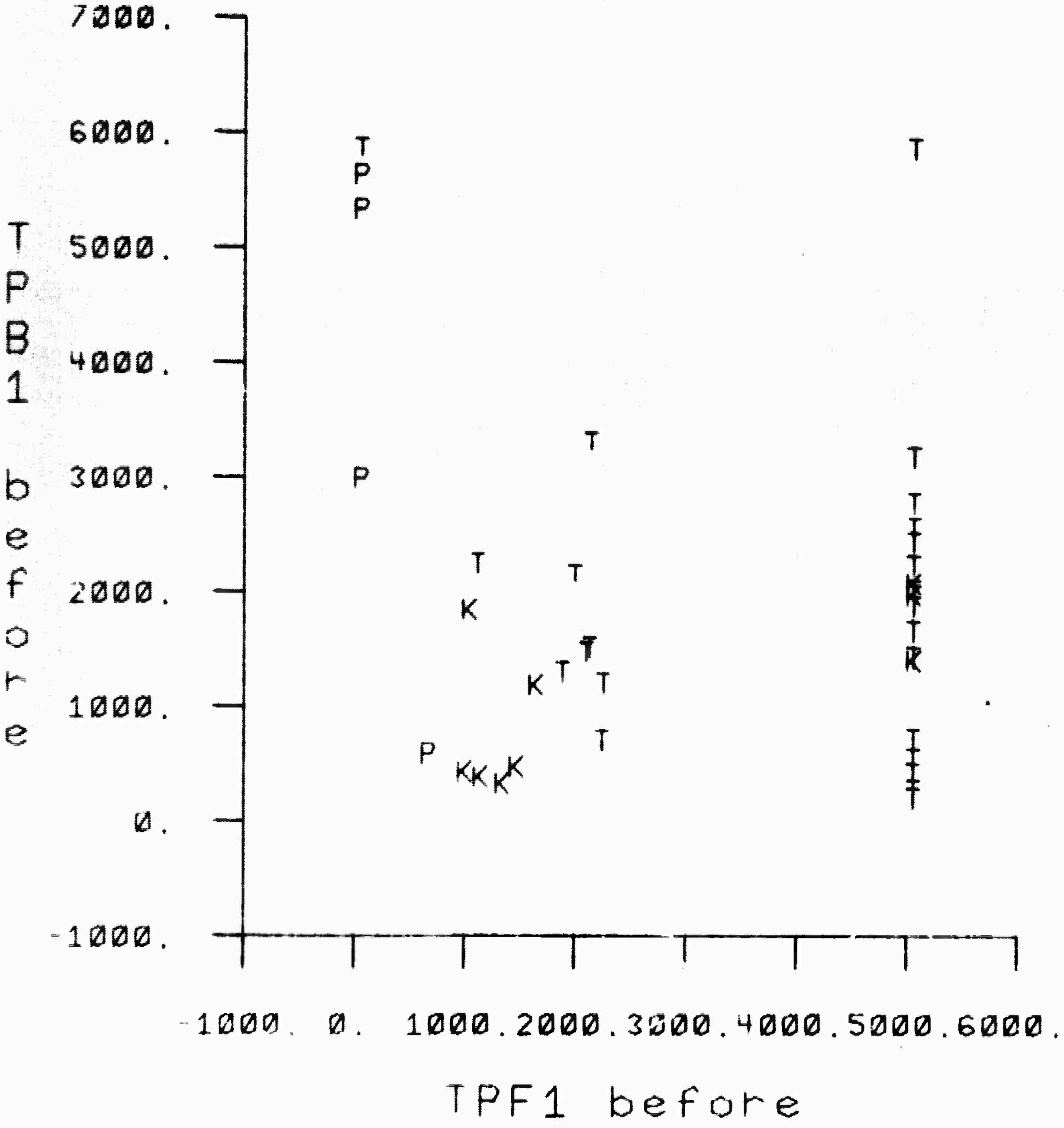

7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
-1000.
TPB1 before
-1000.
0.
1000.
2000.
3000.
4000.
5000.
6000.
TPF1 before
T
P
K

[Figure 4c]



[illegible] before vs TPB1 bef [illegible]

[Figure 4u]

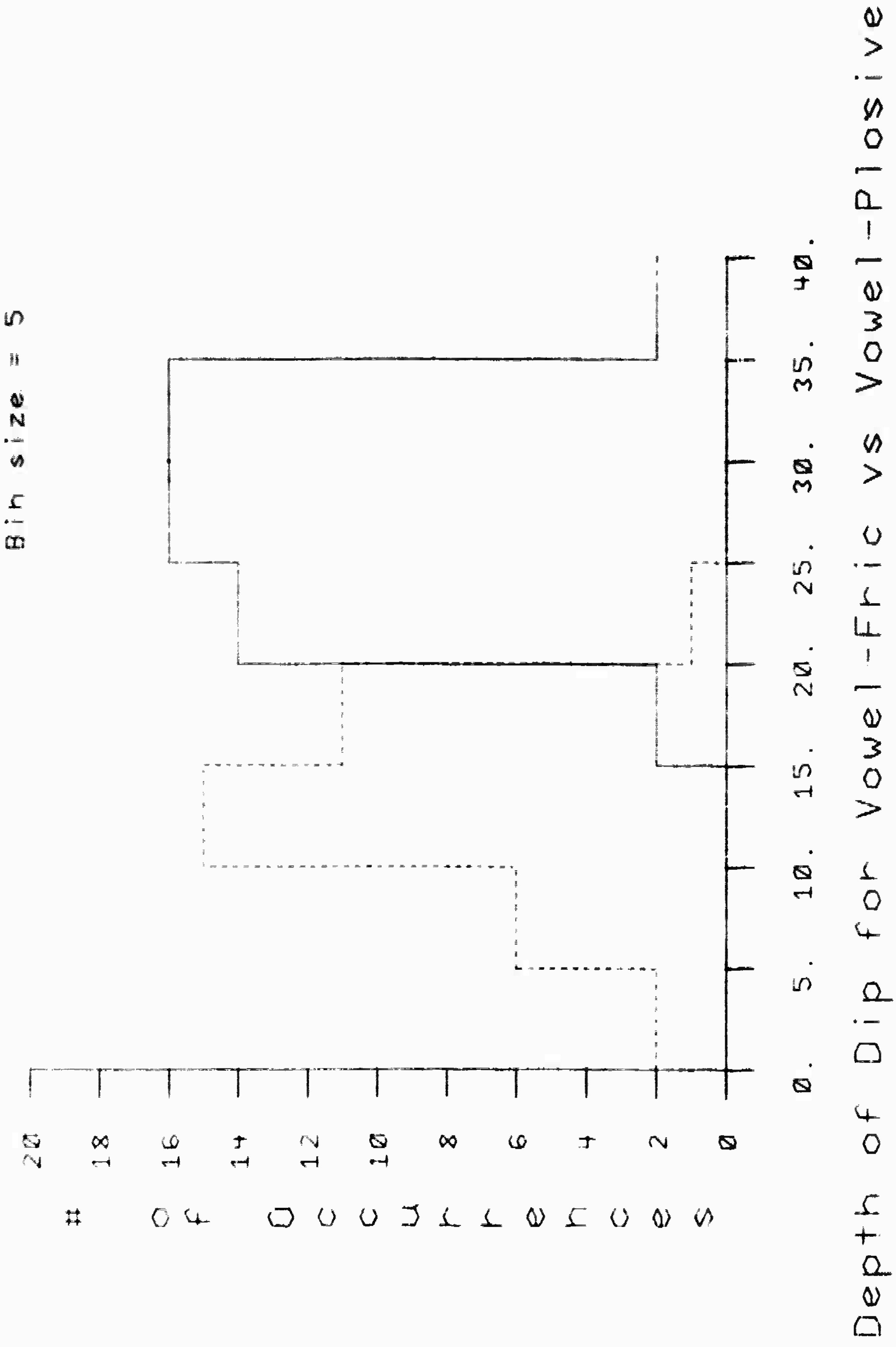

7000.
6000.
5000.
4000.
3000.
2000.
1000.
0.
-1000.
-1000.
0.
1000.
2000.
3000.
4000.
5000.
6000.
TPB1 before
TPF1 before
T
P
K

[Figure 4e]

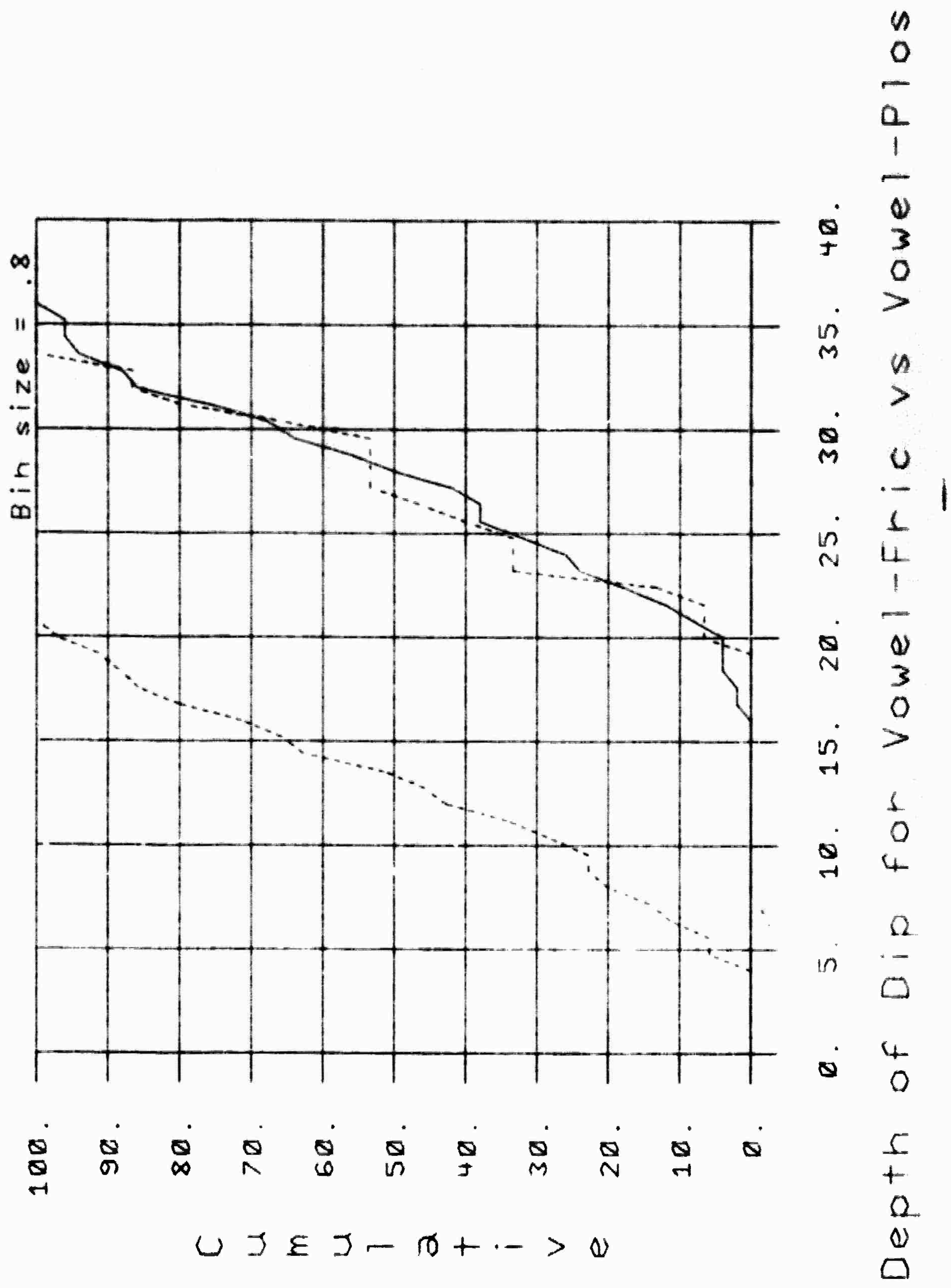
3) For strident fricatives [S,SH,Z,ZH], the distinction of dental [S,Z] vs. palatal [SH,ZH] was made using the two-pole-frequency 2/3 of the way into the fricative. Out of 60 cases used in the statistics program, there were 2 errors: an [S] followed by an [R] was classified as [SH], as was an [S] followed by a [Y]. Paying attention to both the transitions of the peak frequencies and the following context should eliminate these errors.

4) Deciding whether a dip in energy between a vowel-like region and a fricative region was the normal dip expected or rather an indication of a vowel-plosive-fricative, vowel-plosive-aspiration or vowel-affricate sequence was done using the depth of the dip alone. This depth was simply computed as the maximum value of the energy in the preemphasized signal in the vicinity of the dip minus the minimum value. Though this decision was frequently made incorrectly during the human parameter reading experiments described above, the programmed depth criteria performed quite well. Figure 5a compares histograms of th! depth - using a 5 dB bin size - for 37 cases of vowel-fricative (dotted line) and 50 cases of vowel-plosive (solid line). A boundary at 20 dB would result in 3 errors in the 87 cases. The cumulative distributions are shown in Figure 5b, along with a third distribution (dotted line on the right) which represents 4 vowel-affricate sequences and 10 vowel-plosive-fricative sequences (included among the 50). For any samples which fall between 16 and 21 dB - 14 out of 91 do - there would have to be two segmentation paths: vowel-fricative and vowel-plosive, each with a likelihood dependent on the actual depth.

5) In human parameter reading we also found it difficult to decide whether a fricative region between a silence and a vowel-like region represented the aspiration due to an unvoiced plosive, the heavy aspiration due to a [T-R] or [K-R] cluster, or a fricative between a plosive and vowel. 82 examples were separated into these three categories using the duration of the frication and the maximum value of energy from 3400-5000 Hz. during the frication. There were 4 errors. Grouping [R] with other vowels left only 2 errors in the 2 class distinction of plosive-sonorant vs plosive-fricative-sonorant.


Bin size = 5
Depth of Dip for Vowel-Fric vs Vowel-Plosive
# of Occurrences
0
2
4
6
8
10
12
14
16
18
20
0.
5.
10.
15.
20.
25.
30.
35.
40.

[Figure 5a]


Bin size = .8
100.
90.
80.
70.
60.
50.
40.
30.
20.
10.
0.
Cumulative
0.
5.
10.
15.
20.
25.
30.
35.
40.
Depth of Dip for Vowel-Fric vs Vowel-Plos

[Figure 5b]

Work on the above distinctions is far from complete. Those mentioned are given primarily as examples of the type of algorithms developed using the statistics facility.

Richard Schwartz

Victor Zue

References

[1] Makhoul, John and Jared Wolf, "The Use of a Two-Pole Linear Prediction Model in Speech Recognition", BBN Report No. 2537, Bolt Beranek and Newman Inc., Cambridge, Ma. (September 1973).

III. LEXICAL RETRIEVAL

Recent work on the lexical retrieval component has consisted of the formulation, implementation, and extension of our scoring philosophy and lexical lookup procedure, along with corresponding work on our lexicon. Much of this work resembles that done on the CASPERS system [1], with specific extensions to generalize the lookup component to handle segment lattices, probabilistic segment specification, potential scoring, etc. The three areas that will be discussed here will be:

1) Scoring Philosophy

2) Lexical Lookup

3) Phonetic and Phonological Representation

A. Scoring Philosophy

Let U_i be the ith utterance in a enumeration of all acceptable utterances.

Let PM_{ij} be the jth pronunciation model associated with utterance U_i (i.e. an underlying representation of a particular pronunciation of the utterance).

Let F(t) be the acoustic waveform.

Our scoring philosophy is predicated on finding the most probable utterance U_i and pronunciation model PM_{ij}, given the waveform F(t). I.e. Find U_i and PM_{ij} such that $P((U_i,PM_{ij})|F(t))$ is maximized. (This philosophy is discussed in more detail in [1].) Using Bayes Rule we find that:

$$P((U_i,PM_{ij})|F(t)) = P(U_i,PM_{ij})*P(F(t)|(U_i,PM_{ij}))/P(F(t)) \qquad (1)$$

By writing the probability expression in this way, we can more easily isolate its dependence on pragmatics, semantics, syntax, prosodics, and phonetics in a way which is not apparent in the original expression. We do this by noting that the new expression can be broken into three different components:

The first component, $P(U_i,PM_{ij})$, can be written as $P(U_i)*P(PM_{ij}|U_i)$, where $P(U_i)$ is the a priori probability that utterance U_i is spoken, and $P(PM_{ij}|U_i)$ is the probability that pronunciation model PM_{ij} characterizes the acoustics, given that U_i is spoken. The former is determined largely by the syntax, semantics and pragmatics of the task domain, while the latter, though affected by them, is primarily a function of phonetic and prosodic information. (We are assuming that the pronunciation model PM_{ij} characterizes both the phonetic and prosodic information in the acoustic waveform). $P(PM_{ij}|U_i)$ is primarily determined by phonetic implications of U_i through specific word pronunciations (and their predictable word boundary effects) and secondarily by prosodic implications of the syntax, semantics, and pragmatics of U_i.

The second component, $P(F(t)|(U_i,PM_{ij}))$, is the probability that the observed acoustics, $F(t)$, would have been produced, given that utterance U_i was spoken with pronunciation model PM_{ij}. Whatever effect U_i might have had has already been encoded in its pronunciation model PM_{ij} and therefore may be deleted from the expression. Hence $P(F(t)|(U_i,PM_{ij}))$ becomes effectively $P(F(t)|PM_{ij})$ which is purely a function of acoustics and acoustic phonetics.

If $F(t)$ can be partitioned into segments which correspond one-to-one with the phonemes in the model PM_{ij}, we see that $P(F(t)|PM_{ij})$ can be decomposed into a product of probabilities.

$$\begin{aligned} P(F(t)|PM_{ij}) = {} & P(F_1(t)|PM_{ij})* \\ & P(F_2(t)|PM_{ij},F_1(t))* \\ & \ldots* \\ & P(F_n(t)|PM_{ij},F_1(t),\ldots,F_{n-1}(t)) \end{aligned} \tag{2}$$

where PM_{ij} is a sequence of phonetic elements or phonemes $A_{ij}(1)$ $A_{ij}(2)$ ... $A_{ij}(k)$ ... $A_{ij}(n)$, and $F_k(t)$ for $k=1,n$ is the portion of the waveform corresponding to $A_{ij}(k)$.

(We assume that the pronunciation model PM_{ij} has already been adjusted to reflect phonological effects (e.g. at word boundaries), alternate word pronunciations, prosodics etc.) The idea that each phoneme in PM_{ij} "produces" a matching segment of $F(t)$ is implicit in our choice of a sequential pronunciation model. However, the correspondence between $A_{ij}(k)$ and $F_k(t)$ is not quite as simple as we would like.

Let $PM_{ij}(k)$ be $A_{ij}(k)$ in the context of $A_{ij}(1)$ $A_{ij}(2)$ ... $A_{ij}(k-1)$ and $A_{ij}(k+1)$ ... $A_{ij}(n)$.

Then equation (2) can be rewritten as:

$$P(F(t)|PM_{ij}) = P(F_1(t)|PM_{ij}(1))* \\ P(F_2(t)|PM_{ij}(2),F_1(t))* \\ ...* \\ P(F_n(t)|PM_{ij}(n),F_1(t),...,F_{n-1}(t)) \quad (3)$$

since only phoneme $A_{ij}(k)$ (in the context provided by PM_{ij}) is responsible for $F_K(t)$ (assuming correct segmentation).

The third component, P(F(t)) is independent of particular utterances and pronunciation models. Because it is independent it will not affect the ultimate ranking or ordering of any two theories spanning the whole waveform. However, when theories are composed of word sequences which span different portions of the acoustic waveform, the probability of the waveform over these different portions must be known in order to correctly rank each of the theories. In order to see how each portion of F(t) affects the value of P(F(t)), we note that P(F(t)) can also be decomposed into segment size pieces as:

$$P(F(t)) = P(F_1(t))* \\ P(F_2(t)|F_1(t))* \\ ...* \\ P(F_n(t)|F_1(t),...,F_{n-1}(t)) \quad (4)$$

Since we will never be able to exhaustively score every possible utterance, we are forced to search some selected subset of the acceptable utterances, skipping those which appear to be unlikely. We desire therefore to pursue the most likely theories first. This will only be possible if theories spanning different portions of the acoustics can be ranked correctly. It might well be possible to find such most probable utterances without precise calculation of P(F(t)) over different regions of F(t), but we must keep in mind the very real possibility of having to search a space far too large to be practical or possible for a successful real time solution if such ranking is done poorly.

The value of a scoring philosophy, no matter how well formulated, is for all practical purposes only as good as its implementation. We know from experience though, that properly motivated simplifications can be made which permit accurate approximations of the scoring philosophy. Presently two simplifications, each reducing the extent of the dependent context indicated by the scoring philosophy, appear to be most reasonable.

The first simplification results from the observation that while each $PM_{ij}(k)$ may produce a slightly different looking acoustic waveform, the significant waveform characteristics can be accounted for if a single phonetic element, $A_{ij}(k)$, and a small local context is known. Every PM_{ij} can now be rewritten in terms of a finite set of new symbols if each $A_{ij}(k)$ and its relevant local context is represented by a single symbol $B_{ij}(k)$.

$$\text{I.e.}\quad PM_{ij} = B_{ij}(1)\; B_{ij}(2)\; \ldots\; B_{ij}(k)\; \ldots\; B_{ij}(n)$$

$$\text{where } B_{ij}(k) \simeq PM_{ij}(k) \quad k=1,n.$$

Eouation (3) can now be rewritten as:

$$P(F(t)|PM_{ij}) = P(F_1(t)|B_{ij}(1))* \\ P(F_2(t)|B_{ij}(2),F_1(t))* \\ ...* \\ P(F_n(t)|B_{ij}(n),F_1(t),...,F_{n-1}(t)) \qquad (5)$$

The second simplification results from the observation that the conditioning of the probabilities in the above equation on the segments $F_k(t)$ beyond adjacent segments compensates for theoretical differences between the influencing context expected by $B_{ij}(k)$ and the actual context observed. We assume, therefore, that the only differences which are meaningful are those which occur during the region of influencing context encoded by B_{ij} (e.g. one segment on either side).

Hence:

$$P(F(t)|PM_{ij}) = P(F_1(t)|B_{ij}(1))* \\ P(F_2(t)|B_{ij}(2),F_1(t))* \\ ...* \\ P(F_n(t)|B_{ij}(n),F_{n-1}(t)) \qquad (6)$$

Notice that the probability still is very much contextually dependent (i.e. independence assumptions have been made only where independence is well motivated). The change to B_{ij}'s as a means of encoding influencing contextual effects in a unique symbol provides an efficient technique for avoiding the apparent circular necessity to recognize the context of each A_{ij} in order to correctly compensate for its effect.

Much of the analysis so far has been concentrated on the decomposition of $P((U_i,PM_{ij})|F(t))$ into segment size pieces (for the sake of clarity and ease of presentation). In out implementation, however, this probability is computed in word size chunks, based on individual word scores which are in turn based on the scores of a series of the segments "matched" with the B_{ij}'s (contextually compensated A_{ij}'s) of each word. e.g. If in pronunciation model PM_{ij}, a certain word spans the k+1 to k+m segments, its pronunciation model, WPM, is,

$$B_{ij}(k+1)\ B_{ij}(k+2)\ \ldots\ B_{ij}(k+m)$$

and its score is calculated as follows:

$$\text{Word Score} = \prod_{o=1}^{m} \frac{P(F_{k+o}(t)|B_{ij}(k+o),F_{k+o-1}(t))}{P(F_{k+o}(t)|F_1(t),\ldots,F_{k+o-1}(t))} \qquad (7)$$

B. Lexical Lookup

We desire to have a lexical lookup procedure which has the following capabilities:

1) It permits consistent implementation of the scoring philosophy.

2) It is relatively insensitive to random occurrences of noise.

3) It is capable of being extended to handle large vocabularies.

4) It permits alternate pronunciations.

5) It handles missing and extra boundaries (segmentation errors).

6) It handles phonological word boundary effects.

7) It makes accurate compensation in its scoring procedure for effects due to contextual dependence.

8) It operates fast and efficiently.

9) It can work on selected portions of the vocabulary (e.g. due to syntax selection or word length constraints).

During the past quarter implementation of a compiler and extension of the lexical lookup procedure used in CASPERS has been accomplished. What follows is a brief description of how the lexical lookup component works.

In a search of the entire vocabulary, one would not like to (accidently) reject a word until it is known that a acceptably high word score can never be achieved The fact that we assign to each word a score that is built up from the product of the scores of its component segments, together with the fact that it is possible to handle phonological word boundary effects in a efficient manner if all words beginning the same are grouped together, strongly suggest a tree structured vocabulary.

As a result the lexical lookup procedure depends upon a prestructured vocabulary tree. The purpose of the compiler is to assemble a set of words and word boundary rules into an appropriate tree structure [2]. Any path starting at the root and traversing through the tree corresponds to the pronunciation of some word in the vocabulary. The score calculated for the path is, in effect, the score for the associated word. Note however that because many paths are merged together near the root of the tree, the total effort to compute all such word scores is reduced substantially. See Figure 1.

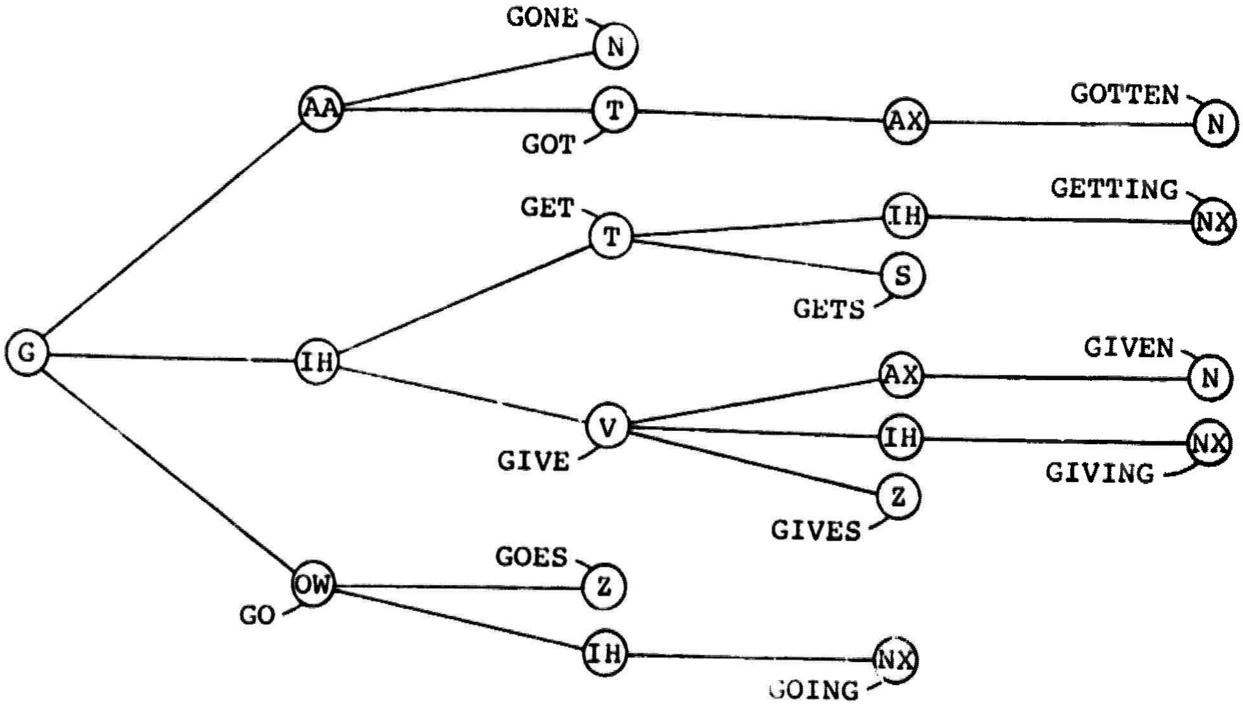


Figure 1: Sample Tree Structure

The set of word scores is computed using a stack of paths (pointers into the tree structure). Each has an associated score for the path from the root to that place in the tree. The stack is updated by taking each such path and its score, stepping one level deeper into the tree, and scoring each subsequent path relative to the old path score. If the score of any particular path through the tree should get sufficiently poor relative to other paths and it is known that pursuing any path in that subtree could not result in a score equal to or exceeding the minimum allowed score (set by a threshold), the path and the subtree under it may be thrown away, thereby saving additional computation. Thus both scoring and rejection are done on sets of words (on subtrees) in an efficient and satisfying manner. Again this technique is discussed in detail in [1].

C. Phonetic and Phonological Representations of the Lexicon

In order to begin some incremental simulation experiments, we have decided to temporarily fix the dictionary for the travel budget management domain in its current state of approximately 450 words. Also toward that end, we have started to specify and store the phonetic representation of each of the words. Phonetic baseforms have been determined for each word, and phonological rules which will derive alternate pronunciations from the baseforms have been collected. (Though most words will have only one phonetic baseform, having pronunciations which cannot all be predicted from a single baseform with reasonable phonological rules will have more.) These alternate pronunciations will be included in the lexicon tree along with the baseforms, and preliminary calculations indicate that this will result in a two- to three-fold increase in its size.

To ensure the correctness of our lexical representations, we plan to get an outside evaluation of the correctness of these baseforms, phonological rules, and marking of syllable boundaries and stress levels from the Speech Communications Research Laboratory. We also will try to determine quantitative information on the relative likelihood of one pronunciation versus another. For example the word "data" is more often pronounced with a flapped, rather than an un-flapped [t]. Quantitative information of this sort will be incorporated into our new lexical retriever and word verifier.

John Klovstad

References

[1] Klovstad, John W., "Computer-Automated Speech PERception System", Ph.D. Thesis, Electrical Engineering Department, M.I.T., (in preparation).

[2] Klovstad, John W. and Lee F. Mondshein, "The CASPERS Linguistic Analysis System", in Proc. of IEEE Symposium on Speech Recognition, Carnegie-Mellon University, pp. 234-240 (April 1974).

IV. THE SYNTACTIC COMPONENT

During the past quarter progress on the syntactic component of SPEECHLIS has been made on both the grammar and the parser.

The grammar has been extended to include a subgrammar for parsing date expressions which occur frequently in discourse concerning travel budgets. Such diverse ways of expressing dates as "July one", "One July", "July first", "Monday, the tenth of April, 1975", and others can now be successfully parsed. Extensive testing of the sentential complement facility of the grammar was also done, and sentences such as, "It costs four hundred dollars to go to California", "Suppose (that) the budget has five thousand dollars", "I have arranged for John to go to Washington") can all be parsed correctly.

In order to handle particle constructions (e.g. "Should a new budget be made up?", "Can we send him out to California before June?", "I need to figure out how much money I have", "Add the costs up"), we found it necessary to make changes to the dictionary as well as the grammar. These dictionary changes involved marking verbs which can take particles and indicating how the features of the verb-particle pair differ from the features of the verb alone. These changes currently await testing. A list of sentences using particles was sent to Wayne Lea at Univac for inclusion in an experiment to test various hypotheses about prosodic cues to syntactic structures since verb-particle pairs seem to have very different prosodic contours than regular verbs and prepositions occurring together.

Several changes were also made to the form of structures produced by the parser. For example, passivization is no longer undone. That is, a sentence like "The money was spent by John," which was formerly parsed into a structure similar to that produced by "John spent the money", now retains the money as the sentential subject and "by John" as a sentence-level prepositional phrase. The reason is that a sentence such as "The money was spent by February cannot be similarily undone, unless some semantic or pragmatic guidance is used to produce "Someone spent the money by February". It was decided that the parser should produce the surface structure for passive utterances along with an indication that the passive voice had been used, and that Semantics would make its case assignments taking the voice of the verb into account.

In all, approximately 40 sentences have been parsed with the current grammar, and have been found to produce structures which are amenable to semantic interpretation. A list of many of these sentences is given in Appendix C, and parsings for some of them are shown in Appendix D.

One change to the grammar which was made in order to effect a significant change in the parser was the inclusion of a weight, currently a small integer, as an additional component of every arc. This weight was originally conceived of as a rough measure of either (a) how likelv the arc is to be taken when the parser is in that state or (b) how much information is likely to be gained from taking this arc, i.e. how likely the parse path including this arc is to be correct. That these two schemes are not equivalent can be seen by the following example. In a given state, say just after the main verb of the sentence has been found, the arc which accepts a particle may be much less likely than the arc which jumps to another state to look for complements. However if a particle which agrees with the verb is found in the input stream at this point, then the particle arc is more likely to be correct.

Since the relative frequencv of arcs from a given state is already reflected to some extent by their ordering within the state, it was decided that the weights would be associated with information content. The actual weight assigned to each arc reflects an intuitive, though experienced, guess.

The parser was modified to employ the weights in the following way. Each configuration created receives a score which is determined by the score on the configuration preceeding it and the weight on the transition between them. In the simplest case, the score of a new configuration is the sum of the score of previous configurations and the weight on the arc between them. Thus the score on a configuration may be considered the score of the parse path terminating on that configuration. If the arc is a PUSH arc, the score of the terminating configuration also depends on the score which is attached to the constituent used by the PUSH arc. Thus if there are several possible constituents in a well-formed substring table at a given point, those which look the best will increase the score of the paths which use them.

The parser then considers a set of the highest-weighted active configurations and tries to extend each of them in turn before selecting a new set. In this way some parallelism is achieved, less likely configurations are not extended, and some of the dangers of depth first processing are avoided.

Madeleine Bates

V. SEMNET - THE NETWORK UTILITY PACKAGE

In the course of constructing the semantic network for travel budget management, we noted several facilities unavailable in our existing formalism and implementation, which nevertheless seemed essential to have. These included such things as the ability to store information about specific arcs and a way for several people to co-operate on the construction of the same semantic network in a reasonable manner. As these seemed to be of general utility, and not confined to networks for speech understanding research, extensive work was done this quarter on extending and improving our semantic network formalism and its implementation. What follows is a description, albeit a brief one, of the current network package, SEMNET. Where features have been modified or extended from earlier versions of the system (documented in [1,2,3]) those features will be noted, along with the reasons for the change.

A. Network Components

Within the SEMNET formalism, there are three types of entities making up a semantic network: nodes links and augments. A _node_ is a place at which information about a conceptual entity is collected and organized. A _link_ is a directed association either between two nodes, or between a node and some information outside the network. A particular node->link->node triple is termed an _arc_, and an _augment_ is a way of associating both network and extra-network information with one or more arcs.

Nodes may correspond to words, objects, events, etc. -- whatever one wants to have treated as a unique conceptual entity. A node may either be named, by associating with it a LISP print name, or be nameless. Independently, it may possess an "ego" which specifies the reason for its existence as a separate entity. For example, there may be one node whose name is "Brick 1" and another whose ego is "Brick 1 as the lintel of Arch 1". Both names and egos are implemented as *properties* of a node, called PNAME and EGO respectively (where a property is one of two types of network links to be discussed next).

Nodes are connected to each other in this formalism via named links, called *relations* if they are two-way connections or *properties* if the connection is in a single direction. Properties may also be used to associate with a node information outside the network, as exemplified by the PNAME and EGO properties mentioned above. The bi-directionality of relations is effected by means of link inverses. That is, when a relation link of type R is established between nodes A and B, so too automatically is a link of type R-inverse between B and A. The semantic network formalism has also been extended to allow one to declare reflexive relations like EQUALS (i.e. ones which are their own inverses) in order to eliminate redundant inverses.

All network links are named, and each linkname has its own associated node in the semantic net. While this may be treated as an invisible implementation decision by the network designer, one may also take advantage of it, as we have in the SPEECHLIS network, as a place to specify facts true of all arcs with a given linkname. For example, it can be used to store the name of the relation's inverse, such logical properties of a link as whether several arcs with that linkname entering a node should be treated as ANDed or ORed and how arcs of that type associate with other arcs, etc. As will be seen in the following example, this need not be exclusively meta-information (i.e. non-conceptual, logical or probabilistic data). As a result, the distinction between "primitive" links and built-up relations that we had previously made, following Shapiro [4], has become blurred. For example, consider the network fragment:

```
101
  PNAME STATE
  KINDS (SOLID)(LIQUID)(GAS)
  NODETYPE (LINKNAME)

102
  PNAME WATER
  FORMS (Water as a solid 103)(Water as a gas 104)
          (Water as a liquid 105)

103
  EGO Water as a solid
  STATE (SOLID)
  FORM/OF (WATER)
  PNAME ICE
```

Here STATE is both a conceptual entity and the name of a relation. Its existence as a conceptual entity (or node) allows us to specify explicitly such information as its possible values.

As a link, it allows us to say that ICE is water in its solid state. (Note in the above example that PNAMEs are printed in upper case, and EGOs in both upper and lower. The terminal nodes of arcs are printed enclosed in parentheses. The values of property links are printed out straight.

Augments provide a way of associating more information than just a linkname with one or more individual arcs in the network. Augments resemble ordinary nodes, except that they serve as a focus for information about particular network arcs rather than about conceptual entities. Several arcs may have the same augment, and some arcs, no augment at all. Arcs lacking augments are termed "simple", while the others are termed "augmented". The association of augment and arc is made explicit within the network and is effected via the property AUGMENT/OF. For example,

```
12
     APRIORI .8
     AUGMENT/OF [concept of spend 14] (agt) (we)
```

would be an augment node associated with the AGT link from the node 14 (whose ego is "concept of spend") to the node whose print name is "we". The converse association between the arc and the augment is built into the internal mechanism of arc implementation and is accessible via the function GETAUG, to be discussed in the next section. What this augment, together with other stored information about the concept of spending, tells us is that while any person or group of people can be the agent of "spend", our estimated probability of its being "we" is 80%.

The impetus to provide such an augment capability was the desire to associate probabilistic information with individual network arcs, for example, the likelihood that concept A will fill the AGENT case of some concept which is fillable by concepts A,B,C or D, or the likelihood that some particular higher concept is being discussed when word A is spoken. However, other A.I. projects at BBN have adopted this formalism and are finding other uses for these augments, such as SCHOLAR's use of them in implementing I-tags.

B. Implementation

The actual data structure in which a semantic network is stored in the current SEMNET formalism is a LISP array, with each node corresponding to a single array element. A node is uniquely identified by its position in the array, e.g. item 1, item 2, etc, where this integer is called the node's SREF (for semantic referent). Each element of a LISP array can hold two LISP pointers, one of which is used for the list of relational arcs leaving the node, the other for the list of properties. Both of these lists are stored in LISP property list format, a change from our earlier implementation, in order to take advantage of the CONS storage algorithm [5].

As before, all arcs with the same linkname leaving a node are collapsed and stored together for efficiency. Thus the list of relations and the list of properties for the node both have the same form, i.e.:

```
(<linknumber1> (<nodespec>+)
 <linknumber2> (<nodespec>+)
 ...)
```

where <linknumber1> is the SREF of linkname 1 etc., and a <nodespec> is either the SREF of the node at the other end of the link for simple links, or a pair of SREFs for augmented links. In the latter case, the first element is the SREF of the node reached and the second, the SREF of the augment. Each list of nodespecs is sorted by the SREF of the node reached to make for efficient retrieval.

C. Utility Packages

Currently six files make up the SEMNET semantic network utility package. These are:

BASICSEMNET: functions for building and accessing a semantic network

EDITSEMNET: functions for editting a network

PRINTSEMNET: functions for printing a network in readable format

MERGESEMNET: functions for merging two somewhat similar networks

UPDATESEMNET: functions for updating a network created in an earlier version of the formalism.

UTILSEMNET: functions of general utility which are used by the other SEMNET packages and which are not provided for in LISP.

Most of the top-level functions currently in BASICSEMNET, EDITSEMNET and PRINTSEMNET have been well described in [1]. Changes made to them to accomodate the new proplist format for relations and the institution of linknames as network nodes have not changed their appearance to the user. Only the new augment facility has produced changes and additions to SEMNET which differ from the write-up in [1]. These will be discussed in the next section, followed by a description of MERGESEMNET and UPDATESEMNET. (Since UTILSEMNET just contains low-level functions, we will not take the time to discuss it here.)

D. The Augment Facility

Augments can be specified in several ways and at several times during the construction of a network. One can specify the augment: 1) directly, as an argument to such arc building functions as ADDREL, ICONNECT and PUTLINK (see [1] for a description of these and other functions not described herein); 2) in a relation specification (RELSPEC) in a call to a node-building functions like IBUILD or ADDITEM; or 3) later, in a call to AUGLINK, a new function which changes simple links into augmented ones.

The form of the augment information is the same for all of the above:

```
<AUGMENT> :=: NIL (* create a simple link)
                   -> (* create an augmented link in the forward
                      direction.  That is, create a new node
                      (arc node) and set it to point to the
                      given link.)
                   <- (* Do the same for the reverse direction)
                   T (* Make both links augmented.)
                   (-> <AUGINFO>+) (* Create a forward augment
                      and hang off it the information in
                      AUGINFO.)
                   (<- <AUGINFO>+) (* Do the same for the
                      reverse direction.)
                   (-> @ <NODESPEC>) (* Make the node specified
                      in NODESPEC the arc node.)
                   (<- @ <NODESPEC>) (* Do the same for the
                      reverse direction.)
                   ((-> <AUGINFO>+)(<- <AUGINFO>+)) (* Augment
                      both forward and reverse links as
                      indicated. Here again <AUGINFO>+ may be
                      replaced by the sequence @ <NODESPEC>.)
      <AUGINFO> :=:   (<REL> <TERM>) | (<PROP> <VALUE>) (* i.e.
                      a RELSPEC)
      <NODESPEC> :=: <NODE> (* i.e. an integer) | a function
                      which evaluates to a node
```

There are top-level functions for getting an augment, adding information to an augment, editing an augment, deleting information therein, converting an augmented arc to a simple one, and printing an augment. Note that we have enabled only relations and not properties to be augmented, though should the need be felt, the facility could be so extended. The following describes both new top-level functions and changes to existing ones which enable augments to be added and used.

1. Arc Building Functions

(ADDREL ITEMA R ITEMB AUGMENT)

where ITEMA and ITEMB are nodes and R is a relation which has an inverse. ADDREL behaves as before if AUGMENT is NIL. That is, it adds ITEMB to the list of nodes reached by following R links from ITEMA and adds ITEMA to the list of R-inverse links leaving ITEMB. If AUGMENT is ->, T, (-> ...) or ((-> ...)(<- ...)), it creates an appropriate arc node and adds (ITEMB . arcnode) to the list of R links leaving ITEMA. If AUGMENT is <-, T, (<- ...) or ((-> ...)(<- ...)), it again creates an appropriate arc node and adds (ITEMA . arcnode) to the list of R-inverse links leaving ITEMB. e.g. ADDREL(FRUIT KINDS BANANA (-> (APRIORI .4)))

(PUTLINK ITEMA R ITEMB AUGMENT)

where ITEMA and ITEMB are again nodes and R is a relation. PUTLINK behaves as before if AUGMENT is NIL. Otherwise, it adds an augmented link with the appropriate information. Note the only sensible values for AUGMENT here are NIL, ->, and (-> ...), since PUTLINK only creates a link in the forward direction.

(ICONNECT ITEMA R ITEMB AUGMENT)

ICONNECT behaves just like ADDREL, except that ITEMA and ITEMB can be either pnames or forms that evaluate to a list of nodes.

2. Node Building Functions

(IBUILD RELSPEC+)

where RELSPEC :=: (R ITEM AUGMENT), R is a relation, and ITEM is either a node, a pname or a form that evaluates to a list of nodes. IBUILD behaves as before, except that when the AUGMENT in a RELSPEC is non-NIL, it creates the appropriate kind and number of augmented links. For example,

```
(IBUILD (PNAME FRUIT)(KINDS APPLE ((-> (APRIORI .8))
(<- (APRIORI .4))))(KINDS PEAR ->)(KINDS BANANA)
(KINDS QUINCE T))
```

(ADDITEM ITEMA (SUPERELSPEC)+)

where ITEMA is either a node, a pname, or a form that evaluates to a list of nodes, and

```
SUPERELSPEC :=: (R LINKSPEC+)
LINKSPEC :=: ITEM | (+ ITEM AUGMENT)
```

The only difference between this and the earlier version is that one can now specify an augment in a linkspec. If ITEM is a form, the same AUGMENT will be put on the link from ITEMA to each node resulting from evaluating ITEMB.

3. New Functions

(ADDAUGINFO ITEMA R ITEMB AUGINFO)

ADDAUGINFO adds further information to the appropriate arc node. AUGINFO is a list of RELSPECS, as in a call to IBUILD. (Also see definition of AUGINFO above.)

(AUGLINK ITEMA R ITEMB AUGMENT)

AUGLINK changes a simple link into an augmented link. Note that AUGLINK only changes the links specified, and not its inverse. The only sensible values of AUGMENT then are -> and (-> ...).

(GETAUG ITEMA R ITEMB)

GETAUG returns the augment associated with the arc from ITEMA to ITEMB via relation RR if one exists, otherwise NIL.

(IEDITAUGP ITEMA R ITEMB)

IEDITAUGP allows one to edit the property information hung off the arc node associated with the particular link from A to B via R.

(IEDITAUGR ITEMA R ITEMB)

IEDITAUGR allows one to edit the relation information hung off the arc node associated with the particular link from A to B via R.

(REMAUG ITEMA R ITEMB)

REMAUG changes the augmented link from ITEMA to ITEMB into a simple one. It is the reverse of AUGLINK. The abandoned augment is put on the FREELIST for re-use if there are no other arcs with which it is associated.

(DAUG ITEMA R ITEMB)

DAUG, for "describe augment", prints out the arc node associated with the link between ITEMA and ITEMB via R.

E. MERGESEMNET

MERGESEMNET is a package of functions for merging two "somewhat similar" semantic networks, thereby enabling two or more people to work independently on the same semantic network and later combine their results. The merger is invoked by the function MERGENETS, whose two arguments name the two files containing the networks to be merged and whose result is a file containing the merged network, e.g. (MERGENETS <WARNOCK>MYNET <AIELLO>MYNET). The following assumptions are made by MERGENETS:

1. Both semantic networks have been made using only the functions in BASICSEMNET and EDITSEMNET. (i.e. There are no relations, nodes, links, or properties unnatural to the structure building and modifying functions found in these files.
2. Both networks have been filed using the NET: macro found on BASICSEMNET.
3. BASICSEMNET has been loaded into the system in which the merger is being done. MERGENETS also requires UTILSEMNET to be loaded.
4. Relations defined in both networks have the same definitions (i.e. the same inverse), though both networks need not have the same set of relations.
5. Networks may contain augmented as well as simple links. There is one caution however: if the link from node A to node B is augmented in both networks, a message to that effect will be printed out to the user, but only the augment from the first network (i.e. the first file name) will appear in the resulting merged network. It has been left to the user to decide what should be done with possibly

dissimilar and/or conflicting augments.

6. MERGENETS is undoable. However, the fact that the files containing the two input semantic networks remain around and untouched following the merger.
7. The file containing the output of MERGENETS will be a later version of the first argument file to MERGENETS. The output of MERGENETS remains in-core as well for further additions, modifications, or disembowelments.

F. UPDATESEMNET

A set of functions, called by the function UPDATE, exists for bringing semantic networks whose format reflects an older version of BASICSEMNET into the new formalism. It takes as input the name of a file containing an old-format semantic network and outputs a new version of that file containing an up-to-date net. Since UPDATE itself checks the form of the input network, no further specifications need be given by the user on what kinds of updating must be done.

Bonnie Nash-Webber

References

[1] Brown, J.S., R.R. Burton and A.G. Bell, "SOPHIE: Final Report", BBN Report No. 2790 (March 1974).

[2] "Natural Communications with Computers IV: Quarterly Progress Report 9", BBN Report No. 2501 (January 1973).

[3] Woods, W.A., Bates, M.A., Bruce, B.C., Colarusso, J.J., Cook, C.C., Gould, L., Grabel, D.L., Makhoul, J.I., Nash-Webber, B.L., Schwartz, R.M., Wolf, J.J., "Natural Communication with Computers Final Report - Volume I: Speech Understanding Research at BBN", BBN Report No. 2976 (December 1974).

[4] Shapiro, S.C. "A Network Structure for Semantic Information Storage, Deduction and Retrieval", Proceedings of the Second IJCAI, pp. 512-523 (1971).

[5] Teitelman, "INTERLISP Reference Manual," (October 1974).

Appendix A

Digitized Sentences for Travel Budget Task

100. Give me a list of the remaining trips and their estimated costs.
101. What do we have budgeted for the ACL meeting?
102. What is the total budget figure?
103. What trips have been taken since January?
104. List all trips already taken.
105. Change the cost of a trip to Amherst to sixteen dollars.
106. List all trips to California this year.
107. How many trips has Craig taken?
108. What is the round trip fare to Pittsburgh?
109. Is two hundred dollars enough for a four day trip to New York?
110. What is the registration fee?
111. When did Bill go to Washington?
112. I need to take a trip to Los Angeles.
113. Is John scheduled to go to Carnegie?
114. Who paid for my trip to IJCAI?
115. Give me a breakdown of the expense to send one person to London.
116. Change the travel estimate to ten dollars for the bus.
117. The final cost of the trip was fifty-six dollars and sixty-six cents.
118. How much did we ask for?
119. Who´s going to IFIP?
120. How much do we have left in the budget?
121. How much does it cost to send someone to California for a week?
122. Which conference is the most expensive?
123. I want to know what trips Bill will take this winter.
124. Am I going anywhere in late November?
125. When is the next ASA meeting?
126. How much have we already spent?
127. Can we afford an additional person to the ASA meeting in St. Louis?

Appendix B

Example of Ideal Hand Labels

```
1760   *
1760/GIVE
1760 G
1900 IH 2
2920 V
3250   *
3250/ME
3250 M
3840 IY 1
5100   *
5100/A
5100 AX
5470   *
5470/LIST
5470 L
6200 IH 2
7000   *
7000 S
7800 SI
8160 T
8440/OF
8440 AX
8900 V
9220   *
9220/THE
9220 DH
9770 AX
10100   *
10100/REMAINING
10100 R
10520 IY 0
11360   *
11360 M
12020 EY 2
13380   *
13380 N
13680 IH 1
14380 NX
14900   *
14900/TRIPS
14900 SI
15300 T
16400 R
16750 IH 2
17550 SI
18200 P
18350   *
18350 S
19300/AND
19300 EH 1
20300 N
20920   *
20920/THEIR
20920 DH
21160 EH 1
21780   *
21780 R
22400/ESTIMATED
22400 EH 2
23370   *
23370 S
23860 SI
24370 T
24670 IX
24970   *
24970 M
25540 EY 1
26100   *
26100 Y
26440 IX
27500 URD
27900   *
27900/COSTS
27900 SI
28300 K
28980 AO 2
31200 S
32400 SI
33170 T
33450 S
35000/(END)
```

Appendix C: Sample Sentences

Some of the Sentences Parsed by the SPEECHLIS Parser

Monday April tenth.
Monday the tenth of April.
April tenth.
One July.
July one.
April one seventy five. (i.e. April 1, ´75)
Monday the tenth of April nineteen seventy five.
July one nineteen seventy four.
Thirty one April seventy five.
April seventy five.
April nineteen seventy five.
April.
The tenth of April nineteen seventy five.
When is John going?
Who is going to IFIP?
It costs four hundred dollars to go to California.
I want John to go.
We started to spend money.
I want to go.
Suppose that the budget has five K dollars.
I have arranged for John to go.
I arranged that John will go.
Twenty one people.
The trips that were taken in July.
Schedule John a trip to California.
The budgets which have money.
Nine people.
Which is the biggest trip?
Which conference is the biggest?
Give me a list of the remaining trips with the estimated costs.
The trip was taken by Bill.
I want you to cancel that trip.
How much did we spend?
The person to whom I sent money.
The registration fee for that meeting is forty dollars.
Nine people will be going to Pittsburgh in April for the IFIP conference.

Appendix D: Sample Parsings

```
SENTENCE: (APRIL TENTH)

37 CONFIGS, 32 TRANS

  S NPU

    NP DATE NUM  10

            MONTH APRIL

                        ---------------


SENTENCE: (MONDAY THE TENTH OF APRIL)

52 CONFIGS, 46 TRANS

  S NPU

    NP DATE DAY MONDAY

            NUM 10

            MONTH APRIL

                        ---------------


SENTENCE: (APRIL ONE SEVENTY FIVE)

56 CONFIGS, 60 TRANS

  S NPU

    NP DATE NUM 1

            MONTH APRIL

            YEAR 75


                        ---------------

SENTENCE: (THIRTY ONE APRIL NINETEEN SEVENTY FIVE)

115 CONFIGS, 133 TRANS
```

```
  S NPU
    NP DATE NUM 31
            MONTH APRIL
            YEAR 1975
                         ---------------

SENTENCE: (I WANT TO GO)
69 CONFIGS, 64 TRANS
  S DCL
    NP DET
       PRO I
       FEATS NU SG
             ROLE SUBJ
    AUX TNS PRESENT
        VOICE ACTIVE
    VP V WANT
       NP S TOCOMP
            NP DET
               PRO I
               FEATS NU SG
                     ROLE SUBJ
            AUX TNS NIL
                VOICE ACTIVE
            VP V GO
                         ---------------


SENTENCE: (SCHEDULE JOHN A TRIP TO CALIFORNIA)
```

```
194 CONFIGS, 179 TRANS
  S IMP
    NP DET
       PRO YOU
       FEATS NU SG
    AUX TNS PRESENT
        VOICE ACTIVE
    VP V SCHEDULE
       NP DET ART A
          N TRIP
          FEATS NU SG
       PP PREP FOR
          NP DET
             NPR JOHN
             FEATS NU SG
       PP PREP TO
          NP DET
             NPR CALIFORNIA
             FEATS NU SG
                        ---------------
  S IMP
    NP DET
       PRO YOU
       FEATS NU SG
    AUX TNS PRESENT
        VOICE ACTIVE
```

```
     VP V SCHEDULE
        NP DET ART A
           N TRIP
           PP PREP TO
              NP DET
                 NPR CALIFORNIA
                 FEATS NU SG
           FEATS NU SG
        PP PREP FOR
           NP DET
              NPR JOHN
              FEATS NU SG
  (*Two parsings are found in parallel, with the ambiguity
    to be resolved later by Semantics.)
                          ---------------
SENTENCE: (WHICH IS THE BIG -EST TRIP)
88 CONFIGS, 79 TRANS
  S Q
    NP DET ART THE
           BIG
       ADJ SUPERLATIVE
       N TRIP
       FEATS NU SG
    AUX TNS PRESENT
        VOICE ACTIVE
    VP V BE
```

```
      NP N WHQ

         FEATS NU SG

                                   ---------------

SENTENCE: (WHICH CONFERENCE IS THE BIG -EST)

99 CONFIGS, 80 TRANS

  S Q

    NP DET ART THE

           BIG

       ADJ SUPERLATIVE

       PRO ONE

       FEATS NU SG

    AUX TNS PRESENT

        VOICE ACTIVE

    VP V BE

       NP DET WHICHQ

          N CONFERENCE

          FEATS NU SG

                                   ---------------

SENTENCE: (I WANT YOU TO CANCEL THAT TRIP)

152 CONFIGS, 149 TRANS

  S DCL

    NP DET

       PRO I

       FEATS NU SG

             ROLE SUBJ

    AUX TNS PRESENT
```

```
        VOICE ACTIVE
    VP V WANT
       NP S TOCOMP
            NP DET
               PRO YOU
               FEATS NU SG/PL
            AUX TNS NIL
                VOICE ACTIVE
            VP V CANCEL
               NP DET ART THAT
                  N TRIP
                  FEATS NU SG
                                     ---------------
SENTENCE: (THE TRIP WAS TAKEN BY BILL)
131 CONFIGS, 104 TRANS
  S DCL
    NP DET ART THE
       N TRIP
       FEATS NU SG
    AUX TNS PAST
        VOICE PASSIVE
    VP V TAKE
       PP PREP BY
          NP DET
             NPR BILL
             FEATS NU SG
```

```
                    ---------------
SENTENCE: (TWENTY ONE PEOPLE)
42 CONFIGS, 37 TRANS
  S NPU
    NP DET POSTART INTEGER 21
       N PERSON
       FEATS NU PL
                    ---------------
SENTENCE: (I HAVE ARRANGE -D FOR JOHN TO GO)
140 CONFIGS, 126 TRANS
  S DCL
    NP DET
       PRO I
       FEATS NU SG
             ROLE SUBJ
    AUX TNS PRESENT
            PERFECT
        VOICE ACTIVE
    VP V ARRANGE
       NP S FORCOMP
            NP DET
               NPR JOHN
               FEATS NU SG
            AUX TNS NIL
                VOICE ACTIVE
            VP V GO
```